Kinematic morphology of space systems

Thesis

Thesis

KINEMATIC MORPHOLOGY

OF

SPACE SYSTEMS

Volume 1

For the degree of Doctor of Philosophy (PhD)

Open University

Department of Design and Innovation

Faculty of Technology

Submitted By: Eur.Ing. John D. Hobbs M.R.Ae.S. C.Eng.
MSc (Aircraft Design, Cranfield, 1973)

Supervised By: Dr. J. Rooney, Open University

Conducted during the period, October 1997 to March 2007, and

Submitted: 23 March 2007

Kinematic Morphology of Space Systems

Abstract

This thesis considers the robustness to faults of mechanical kinematic systems typical of the type applied to the locomotion sub-systems of planetary exploration vehicles. It is argued that, whereas the electronic, software and control methodologies for such kinematic systems have received extensive attention, the development of the theory supporting the corresponding mechanical architectures has not received the same level of attention.

An introduction to the space systems context of the topic is provided, and used to illustrate the nature of the requirements that evolve for such missions, concentrating on aspects of 'terrainability' – the suitability of a vehicle to manoeuvre on rough planetary surfaces. An approach is investigated which takes concepts from graph theory and linear algebra, and uses these to establish a means for representing kinematic topologies, and, in particular, the 'fault graph' structures, and 'fault classes' that result from the progressive application of faults to nominal kinematic system configurations. Ways whereby the eigenvalues and eigenvectors of the characteristic polynomials of the interchange graph adjacency matrices of various kinematic systems can be applied to represent such systems under nominal and fault conditions are investigated, including development of the 'constraints matrix'. Additionally, the relevance of entropy within a fault graph context is considered, and also techniques suggested for analysing systems in a way which allows a richer representation of the underlying kinematic structure.

Various metrics are considered and a means is established whereby a selection of parameters representing some aspects of kinematic systems' behaviour is used in conjunction with these to provide a means of comparing system configurations with each other, in terms of several 'inter-system distance' measures.

Some success was achieved – 'inter-system distances' were derived for a selection of systems exhibiting different topologies and showed that these can usefully be used to represent some aspects of kinematic system topologies. Some evidence was obtained that it is possible to discriminate between tree-based and looped systems using this method.

Statement of what material has been published previously

The original problem of the robustness of mechanical systems to faults was first presented in 1996 at the European Space Agency (ESA) in the paper 'Achieving Improved Mission Robustness' [25]. Graph Theory as a suitable tool for application to the problem of mechanical robustness was discussed in the paper 'Towards Kinematic Classification Schemes for Planetary Surface Locomotion Systems' [56].

A more detailed description of the application of Graph Theory modelling to (then current) planetary lander concepts was illustrated in a paper presented at Cambridge in 2002 'Kinematic Structure for Robust Mechanical Architectures in Robotic Planetary Exploration [27], with associated British National Space Centre (BNSC) sponsored Space Foresight work being described in 'The BNSC Space Foresight Improved Mission Autonomy and Robustness Programme' [28].

The associated Space Research context and relevant programmes were discussed in a research presentation delivered at The Open University in September 2004: 'Autonomous Systems in Space' [29].

Statement of any joint work

This thesis does not contain work contributed by any other persons.

Sponsoring Establishments

This work was sponsored as follows:

October 1997 to November 1998	Matra Marconi Space (MMS), Filton, Bristol, UK
November 1998 to October 2000	Airbus UK, Filton, Bristol, UK
October 2000 to December 2001	Astrium, Stevenage, UK
December 2001 to March 2007	Airbus, Filton, Bristol, UK

For my parents

I would like to express my sincere thanks to my supervisor, Dr. Joe Rooney, who demonstrated remarkable resources of patience and fortitude throughout the entire PhD process, and who provided so much invaluable instruction and guidance.

Thanks, and much more, are also due to my wife Susan, who stoically endured my obsession with this task for so long, to my elder son William for not allowing me to develop any delusions of grandeur, and to my other son Jonathan for spurring me on to finish before I was overtaken.

'Let these words answer

For what is done, not to be done again

May the judgement not be too heavy upon us'

T.S.Elliott, Ash Wednesday

Contents

List of Figures

List of Tables

PART I

BACKGROUND AND BASIC CONCEPTS

Chapter 1
Introduction

Humankind's perspective on space first left the surface of Earth when Wilbur and Orville Wright successfully flew their 'Flyer' at Kittyhawk in the USA on 17th December 1903 [1], took another major step forward in 1961 when Yuri Gagarin first orbited the Earth [40], and again in 1969 when Neil Armstrong walked on the surface of the Moon [60]. Now sights are set on Mars and beyond, and in anticipation of the first manned missions , a number of robotic precursor missions have been executed, with more in the planning stages. For these to properly execute their function, and prepare the way for humans to follow requires that they are reliable, responsive and intelligent. Furnishing robotic space exploration systems with such capabilities has preoccupied mission scientists, engineers and mathematicians for many years, and continues to do so. Significant advances in celestial mechanics, space mission analysis, planetary mapping, electronics, computer systems and machine intelligence have enabled huge increases in space mission capability. Yet, the mechanical enablers have never received the same level of attention as the other disciplines, although the design of space mechanisms and thermal protection systems are valued and specialised branches of engineering science. Additionally, good progress has been made in the development and application of new materials throughout the aerospace industry. Yet much more remains to be done to develop mechanical robotic infrastructures. This includes the development of designs where faults are more fully integrated into the expected operational scenario, rather than unwelcome intrusions, and allowance is made, and possibly, even, advantage taken, of the changes in system character that this precipitates.

This thesis, therefore, addresses the issue of mechanical robustness to faults in planetary robotic systems, identifying novel means for the analysis of the kinematic morphology of planetary robotics and providing new ways of representing and comparing these systems by focussing on

their kinematic structure. It is also shown that these methods are very suitable for computer analysis, and could provide the basis for development of a computer-based design tool.

1.1 Objective

In more detail, the objective of the research is to lay the groundwork for a mathematically based design process for use in developing autonomous planetary exploration vehicle locomotion subsystems with improved robustness to mechanical failure.

With the high level of effort invested in the development of software, firmware and associated supporting computer hardware for autonomous planetary exploration vehicles, a 'technology gap' is opening between these technologies and the mechanical design of the locomotion systems that they are intended to direct. In particular, it is argued that the 'robustness to mechanical failure' of these, relies over-much on established reliability and redundancy methods, and that scope exists for the development of new approaches.

This thesis investigates methods for the representation of kinematic system mechanical fault and failure characteristics, and for identifying systems with particular selections of these. It is argued that successful development of such techniques would be a major enabler for novel design processes aimed at the development of exploration vehicle locomotion subsystems with improved robustness to mechanical failure.

1.2 Description of Thesis Structure

In order to satisfy the stated objective, this thesis establishes a mathematically based scheme for representing and modelling the topological aspects of kinematic structure in a range of mechanical systems typical of those utilised in robotic space and planetary probes. To facilitate comparison of a kinematic system design under development with a set of requirements visualised as a 'goal' system, a means is developed of comparing any two mechanical kinematic systems in terms of sets of kinematic parameters constructed using the representations derived from the modelling scheme. This allows definition of the proximity of any one kinematic design to another within the representation scheme using a novel concept termed 'inter-system distance'. Proximity is defined as the distance between two points representing any two kinematic systems, positioned in a multi-dimensional 'mechanical design space' defined by a set of kinematic parameters, and with those

parameters also being used as location coordinates for the kinematic system representative points in that 'mechanical design space'.

The nature of the 'mechanical design space' depends on the kinematic parameter set chosen. The parameters can be selected to be associated with aspects of the nominal, fault-free, state of the kinematic system. This state can be represented using the graph theoretical concept of the 'interchange graph'. In this, vertices represent the mechanical links in a system, and edges represent the joints connecting these - see Chapter 3. When the 'nominal states' of two mechanical kinematic systems are represented in this way, the 'inter-system distance' between them is a representation of their proximity in 'interchange graph space'.

The modelling of kinematic systems using only descriptions of their fault-free states provides only a partial representation. The 'nominal state' of a kinematic system can degenerate into a number of other, simpler, 'fault states' under the action of one or more faults. For any kinematic system, these 'fault states' are interrelated by a 'fault tree' where each edge represents the action of the fault that created the succeeding state from the preceding state - a 'fault path' - and each vertex represents the interchange graph of one state of the kinematic system. Thus, the 'fault tree' can be regarded as a 'graph of graphs', with any kinematic system being represented as the totality of its 'nominal state' and all of its 'fault states' – that is all of those system states derived from the original 'nominal state' through the action of one or more faults. When two mechanical kinematic systems are represented using their 'fault graphs', then the 'inter-system distance' between them is a representation of their proximity in 'graph of graphs space'.

The states that lie on the 'fault paths' are shown to fall into one or more 'fault classes' that define the way in which a particular state may either degrade further spontaneously, or under the action of additional faults. This concept is of particular relevance because it has the potential to permit the definition of fault mitigation strategies in terms of transitions between particular 'fault classes', and these strategies may allow the definition of standard mitigation methods in instances when specific combinations of 'fault class' arise.

As a consequence of the specific configurations imposed by fault onset, some 'fault classes' cannot be used for such fault mitigation strategies, and are, effectively, dead ends. Again, this is of

particular relevance in understanding the robustness of any particular kinematic system to mechanical faults, and its suitability, or otherwise, for the application of fault mitigation techniques.

As a means of examining, explaining, and developing the 'fault path' and 'inter-system distance' lines of reasoning, the thesis investigates a number of topics which together allow construction of a coherent treatment of the robustness to failure of kinematic systems. Some of the chapters develop the overall analysis methodology, and some discuss and develop the necessary 'tools' required to implement the proposed methodology. The nature of the problem dictates that these different strands of the problem are developed in parallel, and are finally drawn together in the later chapters to demonstrate their significance within the overall strategic scheme of 'inter-system distance' measurement.

Following this section, **Chapter 1** concludes with a brief resumé of some relevant academic and industrial work that has shaped and defined the area of investigation, and from which the methods and 'tools' discussed in this thesis have been developed. In order to provide an insight into the nature of the robustness to faults problem which is being addressed, **Chapter 2** continues with a summary history of space missions. This discusses the role of autonomy in space missions, and the forms that it can take, how requirements applicable to planetary surface exploration systems can be formulated, and the types of vehicle 'locomotion subsystem' that may result.

These topics are very suitable for treatment using graph theoretical and linear algebra methods, since these may be used to represent the kinematic topology typical of the planetary surface exploration systems being investigated. **Chapter 3**, therefore, previews those particular aspects of these methods which are applied in the thesis, and discusses in some depth their application to the representation of driven joints. This discussion identifies that representation of driven joints and joint failure modes can only be successfully achieved if the modelling technique developed embodies a representation of the number of degrees of freedom in the joints. This leads to the proposition that a 'constraints matrix' be defined, based on the same method of representing kinematic systems as an adjacency matrix based on the system's interchange graph, but where the edges are represented not by '1s', but by an integer representing the number of degrees of freedom

existing between the graph vertices in question. The 'constraints matrix' concept is a key theme in the thesis.

Chapter 4 provides a brief discussion of the 'mechanical design space' – that is, the space within which the kinematic systems being examined are considered to be represented. An understanding of this issue is necessary, since the kinematic systems being examined must be represented within the same space if the concept of 'inter-system distance' is to be applied in any meaningful way. The space is considered to contain representations of all kinematic systems, including those of the system or systems representing the target or 'goal' requirements. To fulfil the latter objective, the concepts of 'goal region' and 'goal point' are introduced as a means of representing the kinematic characteristics of a system most nearly fulfilling any defined mission's requirements. Chapter 4 provides an initial introduction to the topic to allow development of the thesis arguments to proceed, and the subject is expanded upon later, in Chapter 10.

A central theme of the thesis is the representation of the relationship between all of the system states into which any one initial 'nominal' kinematic system can degrade under the action of faults. **Chapter 5** presents some important arguments in the development of techniques for dealing with this topic. Key amongst these is the introduction of the concept of a 'fault path' which represents the degeneration of a system's kinematic capabilities as it is subjected to a sequence of faults by depicting the paths existing between the different 'fault states' created for any one kinematic system by fault action. The 'nominal state' of any kinematic system – that is, its fault-free condition – may have the potential to degrade along more than one 'fault path', and this aggregation of 'fault paths' is referred to as a 'fault graph'. What means to adopt for representing these 'fault states' and 'fault paths' is critical, since this underpins the ability to describe system 'proximities' (as discussed above), to describe the means by which one system becomes another as it fails, and to define any potential route(s) by which it may be possible to regain system function. To this end, the chapter discusses how the 'fault path' may be represented by the eigenvalues and eigenvectors of the 'constraints matrix' of the kinematic system's interchange graph, and also introduces how the symbolic expansion of the determinant of the system's constraints matrix may be used to provide a richer insight into the nature of the different 'fault states'. The chapter concludes with a brief discussion of the geometry of the 'fault paths' investigated, since

determination of any underlying geometrical relationships could add significantly to the potential of the method.

Earlier, it was stated that some of the chapters in this thesis should be regarded as developing an overall methodology, and some as developing the necessary 'tools' required for implementation of the methodology. Thus, once a means of representing a kinematic system has been chosen, it is desirable / necessary to have some, ideally quantitative, or at least, formal, measures for:

- Deciding on the proximity of any two systems – this uses the concept of 'inter-system distance', discussed above, and developed in detail in Chapter 6
- Tracking the probability of occurrence of any of the possible changes in kinematic topology of one system as it degrades into another under fault action - (covered by Chapter 7), and
- Confirming the uniqueness of the kinematic system representations being proposed for describing kinematic topology – this is discussed in Chapter 9 by reference to the topic of cospectral systems.

Chapter 6, therefore, discusses the topic of linear and quadratic metrics, and uses these to illustrate how the definition of the proximity of two kinematic systems is not absolute, but dependent on the metric chosen. How this affects the representation of actual systems is demonstrated by deriving various metrics for some sample systems from the example database. It is important to recognise the significance of this topic which underlies much of the demonstration of principle undertaken in the later chapters. This is because, although the later demonstration is undertaken using a specific metric, it is necessary to recognise that the results arrived at are but one of many possible views of kinematic system relationships that could be derived.

It has already been observed that the concepts of the 'fault tree' and 'fault paths' are central to the thesis. An important aspect of the 'fault tree' of a kinematic system is the probability of particular faults occurring – this determines that 'fault path' along which the system degrades. **Chapter 7,** therefore, introduces the work of Claude Shannon in developing the application of entropy principles to communication theory, and shows how this can, in turn, be used as a means of representing system degradation along a 'fault path' within a 'fault tree' in entropy terms. Entropy is subsequently chosen as one of the parameters used to define the 'mechanical design space', and

is the only parameter embedding the key topic of probability, which is used within the final parameter set defining the 'mechanical design space'.

Chapter 8 is a further critically important chapter because it is here that the 'fault path' concept is examined in greater depth, and its validity for treating 'real' mechanical kinematic systems is established. The chapter opens by demonstrating the relationship between the symbolically expanded determinant of a kinematic system's 'constraints matrix', and the corresponding characteristic polynomial coefficients, and showing how such constraints matrix-derived representations demonstrate the same capacity for analysis using a 'coefficients by inspection' method as representations based on the adjacency matrix. To demonstrate this is important since introduction of the 'constraints matrix' would be of limited value if, by doing so, the attractive features of adjacency matrix based representations were sacrificed. The chapter continues by discussing in greater detail the topic of representing system 'fault states'. The 'fault states' of several simple kinematic systems are derived, and the system 'fault digraphs' (which may be regarded simply as an alternative representation of the 'fault tree') are presented. It is shown how the 'fault states' of a kinematic system fall into one of a finite number of different 'fault classes' and that these can be characterised using the symbolic determinant expansion technique developed in Chapter 5. The chapter also develops a symbology for representing determinant terms as an aid to visualisation of the determinant term relationships which are lost as systems degrade. This ability to group 'fault states' into 'fault classes' is of considerable interest, because it allows patterns in the fault-induced degradation of kinematic systems to be identified, in a way that should facilitate the future description and identification of appropriate fault mitigation strategies.

As commented earlier, determining the uniqueness of any proposed mechanical kinematic system representation underpins the validity of the method proposed for the description of 'fault paths' and 'fault classes'. If the method is not capable of handling the properties of cospectral and isomorphic graphs, then its applicability will be limited. **Chapter 9**, therefore, shows that the methods proposed are, on the basis of an initial assessment, valid from this viewpoint by illustrating how the symbolic determinant approach can readily show the difference between two cospectral systems. The implications for the description of isomorphic kinematic systems are only addressed on a cursory basis, and more work remains to be done here.

The final chapters of the thesis undertake a practical demonstration that the methodologies developed are realistic for practical application. **Chapter 10** starts to draw together and make use of the representations and tools discussed earlier. The definition of a 'goal region' and 'goal point' existing within a 'mechanical design space' described by kinematic system parameters is picked up from Chapter 4, and discussed in greater detail. It is shown how a selection of the parameters developed or identified earlier in the thesis can be used to represent this 'mechanical design space', and, in conjunction with a suitable metric (quadratic metric used as illustration), can be used to derive an 'inter-system distance'. It is shown how sub-sets of these parameters, forming projections of the 'mechanical design space', can be used to derive system proximities on the basis of certain groupings of parameters. The chapter continues by describing a number of simple kinematic systems that are to be used in Chapter 11 to demonstrate the technique of describing the proximity of two systems using the 'inter-system distance'.

Chapter 11 gives a detailed illustration of the method of deriving numerical values for 'inter-system distance'. Some detailed issues regarding the application of some of the parameters selected are discussed and tabulations of 'inter-system distance' are derived using the overall 'mechanical design space', and four projections of that space. Results show that discrimination between kinematic systems with different topologies – for example kinematic systems with tree structures, and kinematic systems with looped structures - is possible using this method, although further refinement of the technique is necessary.

Chapter 12 presents a detailed resumé of the thesis, identifying key themes and outcomes, and provides suggestions for further work.

1.3 Academic and other Influences on this Research – an Historical Perspective

Although the Lunakhod 1 and Apollo Lunar Rover activities of the early 1970's were the first publicly visible applications of planetary exploration vehicles, they were themselves the product of much earlier design and development activities. It will be seen that this thesis draws together lines of investigation from several areas, most of which have roots well in the past, but which are now achieving a modern significance. This trait is typified by the work in the USA during the 1950s and 1960s by Becker [7, 8]. Becker undertook pioneering investigations into the fundamental principles

of surface locomotion. His work was primarily aimed at developing theoretical models for the traction of military vehicles, but came to be adopted as one of the foundation stones of reference on the locomotion of modern planetary robotic systems.

Another, but very different, example of relevant work is that by Grey Walter in the late 1940s and early 1950s [73, 74]. Walter experimented with the intelligence aspects of mobile robotic systems at the Burden Institute near Bristol using an electrical 'turtle'. Although Walter was primarily interested in the human condition, the significance of this work as one of the first steps towards the understanding of autonomous systems seems to have been unrecognised at the time.

Working separately, but contemporaneously, Shannon discussed the mathematical representation of information [61], suggesting coding methods and the concept of treating communication or information transfer as analogous to the variations in entropy first elucidated by Clausius (1822-1888) and Boltzmann (1844-1906). Shannon remained engaged in this work over several years, and the quoted reference was reissued in 1962, together with additional contributions from Weaver [75].

A further, distinct thread of relevant work lies with the origins of Graph Theory that can be traced back to the 19th Century. Recent developments are of particular relevance to the work here - in particular, the classification of kinematic chains on the basis of characteristic matrices or polynomials as suggested by Uicker and Raicu in 1974 / 1975. In two papers, they described a new classification scheme for mechanisms based on polynomials [71], and tests for isomorphism using a characteristic matrix [50]. Yan and Hall further developed this method in a 1981 paper [90] that postulated the use of the coefficients of characteristic polynomials in the study of structural analysis and the synthesis of kinematic chains.

Rooney and Earl [53] further developed the representation of kinematic robotic architectures, and the graph theoretical approaches to kinematic chain representations discussed in that paper are directly applicable to the robotic systems under consideration here.

A subsequent paper by Earl and Rooney [14], developed the application further in a discussion of actuation and distribution components, and the representation within interchange graphs of input links and driven joints. Desirable graph / rooted tree morphologies for such systems incorporating

drive mechanisms were discussed. This leads to the view that desirable configurations of kinematic chains are definable for combinations of actuation / distribution components in terms of tree structure, etc, and that system failure modes are capable of illustration by transformations within the graph representation of the mechanism in question from favourable to less favourable diagrams, such as the heightening of trees.

Karnopp and Rosenberg initially developed Bond Graph techniques in the late 1960s. Later developments were published in the early 1980s by Tiernego and Bos, by Zeid and Chung in the early 1990s, and by Favre and Scavarda in 1996 [19]. A comprehensive introduction to the method is provided in the 2000 book by Karnopp, Margolis and Rosenberg [33].

O'Shea and Eisenstadt, 1984 [45], and Nilsson, 1998 [44], provide comprehensive summaries of artificial intelligence concepts and methods, including established search routines, as well as descriptions of the application of propositional and predicate calculus. There are distinct synergies between these topics and the graph theory work undertaken in this thesis. Search routines can be envisaged based on the fault or failure condition network for autonomous systems, and specialist methods based on, for example, predicate and propositional calculus, used to identify acceptable result conditions.

Close to where Grey Walter conducted his experiments, Winfield and Melhuish now lead the research team at the University of the West of England's Intelligent Autonomous Systems Engineering Laboratory, actively progressing the field of distributed systems, investigating emergent behaviour properties of 'swarms' of robots, in collaboration with BAE Systems' Sowerby Research Centre. A strong sub-theme is the integration into autonomous systems of advanced communication subsystems providing the ability for autonomous systems to interface directly with web-based communication protocols.

To present a comprehensive review of the huge volume of research and development relevant to planetary robotics theory and practice carried out between the 1940s and the present day, and the corresponding evolution in robotics hardware and software design is not considered practical here. However, some specific examples are identified as especially relevant:

In the USA, one of the major players in robotics continues to be 'Red' Whittaker of the Field Robotics Centre of Carnegie Mellon University's Robotics Institute. Carnegie Mellon has ongoing major collaborations with, for example, NASA's Jet Propulsion Laboratory (JPL). JPL is a major player in the development of rocker bogie locomotion systems including the 'Rocky' series. The end results of this development series were the Sojourner vehicle placed onto the surface of Mars by NASA's Mars Pathfinder mission in 1997, and the Spirit and Opportunity rover missions of 2003.

In Canada, the Canadian Space Agency takes an active role in the development of space robotics, particularly in collaboration with MD Robotics (formerly SPAR), who were responsible for the International Space Station's robotic arm. Canadian Universities in the field include the University of Waterloo who are active in the kinematics and dynamics of multibody systems, and also the University of Toronto. In Europe, the European Space Agency (ESA) and the French Centre Nationale d'Etudes Spatiales (CNES) have evolved a large and competent network of internal specialists, specialist institutes and companies active in the field. These include the Politecnico di Torino, Dipartimento di Elettronica and Dipartimento di Meccanica, who collaborated with Alenia Aerospazio on the design of Walkie-6, Astrium (formerly Matra Marconi Space) (Toulouse, France), Technospazio (Italy), and Von Hoerner and Sulger.

In Russia, organisations such as VNII Transmash have a long pedigree in the creation of planetary robots, going back as far as Lunakhod, and including the ill-fated Mars-96 'small stations' and penetrators, lost when the mission failed to achieve insertion into interplanetary trajectory due to launcher malfunction during lift-off in November 1996.

In October 1994, the Earth Observation and Science Division of Matra Marconi Space (MMS) (at the time based in Bristol, but subsequently closed and amalgamated as part of Astrium) submitted a proposal [23] to the European Space Agency (ESA) in response to their invitation to tender for a Mobile Instrument Deployment Device [15]. This was followed in September 1995 by an ESA study for a Moon Based / Free Flyer Interferometer Trade-Off (MOFFIT) - Moon Based Option [24]. Work on this study was significant in identifying the particular problems associated with the reliability of the very long sequences of mechanical actions that can be involved in autonomous planetary operations. In order for the deployment of the interferometer that was the subject of this

study to be successful, upwards of 250 separate sequential mechanism operations had to be achieved virtually fault-free.

MMS proposed a way forward capitalising on research by Husbands and Harvey at the University of Sussex, Brighton, UK into evolutionary robotics [22] and a tentative collaboration between the Department of Cognitive and Computing Systems at that university, MMS, and other organisations was established. Based on this, a response to the British National Space Centre (BNSC)-sponsored 'Foresight' activity, entitled 'Improved Mission Autonomy and Robustness' (IMAR) was initiated, and carried on in two phases until 2001. Work at Sussex is presented in a large number of papers [eg 30], and the group continues to investigate the application of evolutionary techniques to both single autonomous systems, and distributed systems. Additionally, extensive work is being put into evolutionary minimal systems realised largely in electronics.

One of the weaknesses of the first phase of IMAR was that the stochastic nature of the evolutionary methods being pursued by Sussex could result in a technique that would have difficulty gaining acceptance within mainstream spacecraft design because of its inherent lack of definition. A major element of this was considered to be the absence of any rigorous mathematical treatment of the subject matter involved. In IMAR-2, Rooney [55] established a mathematical treatment of some of the issues involved, building upon the hierarchy of geometries first suggested by Klein in the 19th Century, and applying this as a tool to outline the basic principles of kinematic and geometric structure in robot systems, and to provide a formal groundwork for dealing with distributed autonomous systems. This work is referred to further in this thesis in Chapter 5.

Work by Gillies and others in 1999 [20], and by Radice and McInnes in 2001 [49], at the University of Glasgow Department of Aerospace Engineering (DAE) paralleled and complemented that at Sussex, but followed a fundamentally different line of research. In IMAR-2 during 2001, McInnes [37] approached the task by starting from the more established, mainstream, space mission design viewpoint. The DAE artificial agent research has been adapted from studies of biological systems, which paralleled Sussex's work. However, unlike the Sussex research, DAE's work was centred on an algorithm-based approach to the problem of action selection in order to optimise a spacecraft's ability to re-sequence its actions to compensate for major hardware failures.

Parallel to the consortium work on IMAR, Astrium was also active in autonomous spacecraft control (for example, work on the PROBA spacecraft with ESA), and in autonomous star tracking, formation flying and constellation control.

Amongst recent developments in the application of Graph Theory to multibody systems is the work carried out by McPhee and others at the University of Waterloo, Canada [39]. This includes work on kinematic and dynamic cutset equations for multibody systems and methods for satisfying these using through and across variables associated with the graph vertices and edges. The work also includes the development of symbolic computer implementations using Maple and in-house Dynaflex software with on-screen generation of multibody systems. The referenced paper also covers the development of system models through synthesis using sub-system representations, and the application of this to mechatronic systems.

Chapter 2
Space Systems Background

Throughout the space exploration, space science and space exploitation communities, the pressures to establish cost effective and reliable planetary exploration missions are intense. The search for water on the Moon and elsewhere, and the geophysical and exobiological characterisation of Mars are key features in the space programmes of Europe, the USA, Russia and Japan. China also has recently declared ambitious plans in the fields of both manned and unmanned missions. In this cost-conscious, and highly competitive environment, the difficulties of running a manned space programme, particularly in terms of cost, the radiation and micro-particle environments, physiological and other hazards mean that precursor, robotic missions currently receive considerable attention. Consequently, as the planetary exploration programme continues to develop and evolve, the objectives for such robotic missions become more complex, and place greater demands upon the reliability and robustness of the autonomous systems created for these tasks [25]. Typical of current programmes where robotic technologies are paramount are NASA's ongoing operation of the Mars Exploration Rovers Spirit and Opportunity [78], and the European Space Agency's planned Aurora [76] programme of Solar System exploration that combines both autonomous and manned missions in an integrated suite of planetary visits.

In the context of robotic technologies, considerable strides have been made in control system design, and in associated software and logic approaches. Algorithmic, state-based, and fuzzy-logic techniques are only some of the main competitors in the search to identify more fit solutions to the problem of controlling and managing planetary exploration with only limited means of human intervention. Furthermore, traditional analytical, expert system and knowledge-based approaches to design and fault analysis now have rivals in the genetic algorithm / neural network techniques espoused by the evolutionary robotics research community [22].

This thesis argues that the level of attention given to control and software issues is such that a situation has arisen where the control system design is frequently far more sophisticated than the associated mechanical architectures, which are insufficiently flexible to take advantage of the latest developments in control techniques. In particular, mechanical systems with sufficient motion or kinematic functionality to tackle advanced autonomous operations, may exhibit inadequate robustness, in the face of the inevitable system faults, to meet the requisite mission goals.

Construction of new design tools to rectify this situation can be thought of as a two-pronged development, which should provide fresh insights into this problem. On the one hand, the establishment of system classification schemes is a step towards the consistent representation of practical examples of autonomous systems. On the other hand, it is necessary to identify methods for the synthesis of theoretical systems within a design space evolved by reference to the capabilities of actual systems. This dual approach should provide valuable building blocks for the future development of novel design techniques.

This thesis considers ways in which the kinematic structure of planetary exploration vehicles can be represented, and how this representation may be used to define those kinematic features which make for successful, robust systems, well fitted to the complex mission goals required. Such a theoretical kinematic classification should facilitate the future foundation of novel, computer based, design tools for the development of new and innovative autonomous systems.

2.1 Overview of Space Systems

2.1.1 Origins

It is commonly agreed that the history of space technologies (and, in due course, of space systems) begins with the invention of gunpowder in about the 10th. Century by the Chinese, and that it was the Chinese who also later invented rocketry. The Russian teacher K.E.Tsiolkovskii is credited with the basic ideas of liquid propellant rockets and staged rocketry in 1903, and the American R.H.Goddard with the first successful firing of a liquid propellant rocket in 1926. The technology was greatly developed at Peenemünde under Wernher von Braun during World War II, resulting in the single stage V2 rocket. It is widely accepted that the launchers that later formed the core of the

early American and Russian space efforts were direct developments from the V2, and attributable directly to the German (and other nationality) scientists captured at the end of hostilities in 1945.

The first true milestones in space research were the launching of the first artificial satellite, Sputnik 1, by the USSR in 1957 and the first manned space flight with the Russian Yuri Gagarin as cosmonaut in 1961. Fuelled by Cold War rivalry, many missions were executed in the 1960s by the USA and the USSR, culminating in the USA's successful landing of Apollo 11 on the Moon on 21st. July 1969 and Neil Armstrong's pronouncement 'That's one small step for (a?) man, one giant leap for mankind'[60].

This success was achieved at the cost of several Russian and American lives, including three Apollo crewmembers killed in a 1967 launch pad fire.

Space technology spans a huge range of activities, including Earth observation satellites, communications satellites, space science missions and space exploration / exploitation activities.

Manned exploration of the Solar System started with the 1969 Apollo 11 Moon landing, followed by Apollos 12, 14, 15, 16 and 17 up until 1972. However, it must be remembered that these landings were the culmination of many American and Russian manned missions beforehand, and also that 105 planetary probe missions, mainly Moon, Venus and Mars fly-bys were carried out before the first Lunar landing. The USSR's Venera 3 was the first probe to reach a planet (Venus) in 1966, although contact was lost before atmospheric entry. In 1975, the USSR landed Venera 9 on Venus, and this returned images to Earth for 53 minutes. In the same year, the USA succeeded in putting Vikings 1 & 2 onto the Martian surface.

2.1.2 Mission Types

Missions fall into a number of distinct categories such as Planetary and Cometary Fly-bys, Deep Space missions, Planetary Orbiters and Mappers, Autonomous or semi-autonomous Planetary and Cometary Landers, and Satellites. In total, 149 planetary probes were launched by 1975, and this number has now increased to well in excess of 200. Recent attention has been focussed on Mars with the July 97 Mars Pathfinder / Sojourner mission, the January 2004 Spirit & Opportunity missions (currently ongoing – November 2006), and ESA's Mars Express, with its failed Beagle 2 lander [11, 32, 34, 62].

A detailed résumé of all this activity is not appropriate here. Highlights include the 1972 Pioneer 10 Jupiter fly-by and the Pioneer 11 Jupiter / Saturn fly-by. The subsequent Voyager spacecraft (see Figure 2.1) were highly successful with Voyager 1 taking 17,000 Jupiter and Saturn images, ending in 1980, and Voyager 2 photographing Uranus & Neptune & planned to continue activity until 2017. (Voyager 2 has now almost escaped the Sun's gravitational influence to become humankind's first interstellar spacecraft).

In 1985, the Giotto probe (see Figure 2.2) made significant contributions to our understanding of comets with fly-bys of comets Halley and Grigg-Skellerup. The Cassini / Huygens mission was launched in 1997 to the Saturnian satellite Titan, and recently returned photographs of Saturn's ring system, and of the surface of Titan.

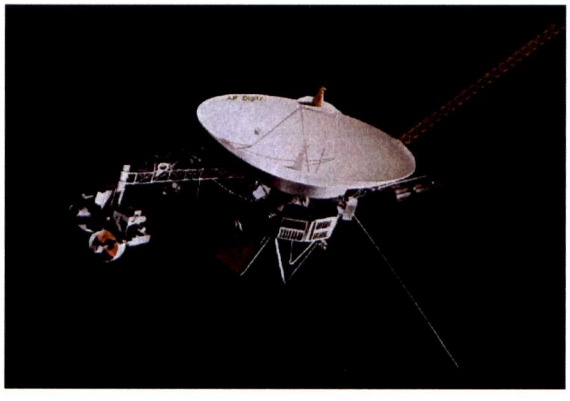


Figure 2.1: Voyager - Deep Space Mission

(NASA image)

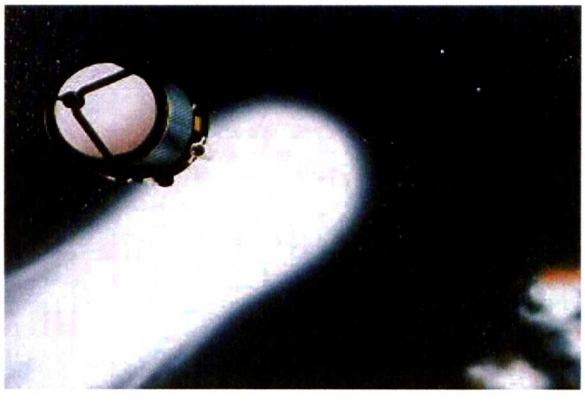


Figure 2.2: Giotto - Cometary Fly-by Mission

(Astrium image)

The value of orbital observatories has been proved with the Hubble Space Telescope, launched in 1990. It was feared that this was in its last phases due to servicing missions having been stopped because of problems with NASA's Shuttle Transportation System. However, in October 2006, NASA announced that there would be one further Shuttle servicing mission in Spring 2008 [41], and this should extend Hubble's operational life by approximately five years. With additional, more powerful orbital observatories also being planned, this raises the interesting prospect of a short period during which there may be more than one powerful observatory in orbit, with all that this implies for collaborative observations and the associated enhanced scientific return.

All the major space agencies have full and challenging inventories of projects leading many years into the future.

2.1.3 Mission Hardware

Because there are many mission / probe types, it is difficult to classify them into any meaningful categories. Typically, a 'traditional' satellite might be approximately 4000 kg, but recently there has been a strong emphasis on reduced cost (and size) missions, although this has many risks, as evidenced by the loss of the Mars Polar Orbiter and Mars Global Surveyor missions. Current Research and Technology trends are towards the introduction of smaller and smaller satellites, the so-called micro-satellites, ranging from one to a few tens of kilograms – progressively employing greater degrees of micro- and nano- technology.

The true so-called nanosatellite - below 1 kg – remains very much a technology in its infancy, but has great potential, for example, in the autonomous servicing of space stations. Large numbers of nanosatellites are proposed as a means of constructing extremely large virtual arrays – that is to say, arrays comprising 'constellations' of separate satellites, constrained to orbit within very tight spatial relationships to each other, and acting in concert to mimic the behaviour of very large, single arrays. (This is an excellent example of the significance of autonomous systems in space science, since the requirements for autonomous navigation and station keeping in this application are very stringent).

2.1.4 Launchers and Space Stations

Launchers: In 1981, NASA introduced its partially reusable Shuttle Transportation System (STS), with five Orbiter Vehicles (OVs), marking the end of the era of dependency on totally expendable rockets. However, for many missions, the use of expendable launchers remained the most common method, particularly when the STS encountered the double blow of losing 7 crew-members when OV-101 Challenger exploded 73 seconds after lift off in 1986 [77], and a further 7 crew-members died when OV-102 Columbia was lost during re-entry on 1st. February 2003 [79, 81].

Currently, activities continue with the development of both partially and completely reusable systems. A number of research projects to develop this technology further have been undertaken, and there is still much activity in this field. NASA's technically promising, but 'financially challenged' X-33 / X34 / Venture Star project, although much heralded at the time, was, unfortunately, cancelled in March 2001 [84]. A number of 'shuttle-type' projects exist, although

mostly these still aim to use the expendable launcher / reusable shuttle approach. Current examples (2006) are the Russian Soyuz-launched Kliper, being studied by RKK Energia at Korolyov, near Moscow [63], NASA's Shuttle replacement – the Orion-launched CEV (Crew Excursion Vehicle)[13, 35, 64], and the sub-orbital VSH (Véhicule Hypersonique Réutilisable) planned by ACE (l'Astronaute Club Européen) and Dassault Aviation [42]. Of these, it has to be said that the CEV is that most likely to be realised. A significant disappointment in this area of technology was the cancellation of ESA's Hermes programme, intended to be launched by Ariane 5, which would have made significant progress in moving forward Europe's launcher capability.

Fully-reusable spaceplanes can be subdivided into those where the entire spaceplane goes into orbit, and those where a carrier aircaft is used to carry the spaceplane to a suitable altitude before release. The former category is typified by the studies carried out for the Bristol Spaceplanes' Ascender by Ashford [6]. Engine technologies are probably more key to successful development of 'Single Stage to Orbit' (SSTO) spaceplanes than is airframe design where, although challenges exist, technological solutions have, in the main, been identified. A good example of SSTO engine research is that currently in hand at Reaction Engines in Culham, Oxfordshire, UK [65], building on work on the RB-545 dual mode engine developed for Hotol.

The British Aerospace Hotol project [86], was originally conceived in 1984 as an SSTO configuration, although, as a means of sidestepping funding issues, an air-launched interim version in collaboration with the CIS was developed. The Reaction Engines' Skylon continued to be developed using Hotol technology until 2003 [79]. More recently, significant success was achieved by Scaled Composites' 'Spaceship One' which has now opened the door to serious consideration of 'space tourism' by operators such as Richard Branson and his proposed 'Virgin Spaceplane' based on Scaled Composites' 'Spaceship Two' [72]. This now looks to be a serious proposition for the commercialisation of space. Whether this is a good or bad thing is debatable, but it seems likely that the commercial incentive is necessary if space exploration is really to move forward in the future.

Space stations: The first Space Station was the USSR's Salyut 1, launched in 1971. First significant stays were aboard the 1973 Skylab programme, which suffered thermal shield damage on take-off, but which was 'jury-rigged' and subsequently occupied for 171 days by three

successive three-man crews. Salyuts 3, 4, 5, 6, and 7 provided significant USSR presence in orbit between 1974 and 1985. This period also saw the 2-day Apollo-Soyuz Test Programme link-up between Apollo and Soyuz capsules in 1975.

Starting in 1986, the USSR's Mir station survived for 15 years before the decision was taken to de-orbit it, and its final re-entry occurred in March 2001. Mir received much, undeserved, poor press coverage, but vastly extended knowledge of the physiological effects of long-term manned space travel. Work now concentrates on the politically sensitive International Space Station (ISS)(see Figure 2.3), incorporating modules from the USA, Russia, Japan and Europe, plus components from other countries.

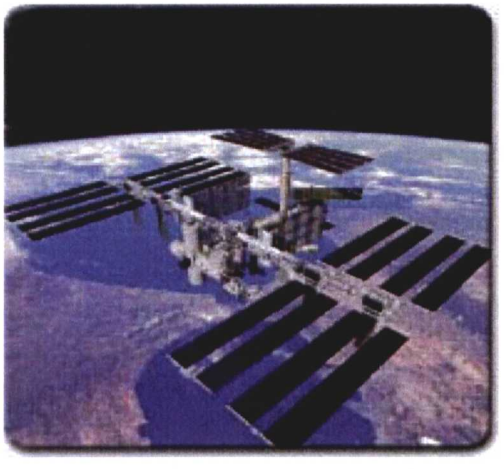


Figure 2.3: International Space Station

(NASA Web Site image)

This work is severely hampered by the ongoing problems with the Shuttle Space Transportation System, the reduced numbers of shuttles creating significant logistical problems with orbiting of new equipment, and with maintenance and supply missions. Without the ongoing significant numbers of Russian Soyuz resupply and crew-exchange missions, the ISS would be severely jeopardised.

2.1.5 Future Plans

Future plans are extensive and complex, but will not be discussed here, given the limited space available. Suffice it to say that these plans include the current NASA and ESA programmes of robotic Mars probes, leading to a manned mission about 2014, plus ongoing comprehensive science and technology missions [84]. During 2005 and 2006, both China and the USA [51] sign-posted a possible human return to the Moon for longer stays than with Apollo. As mentioned previously, sub-orbital tourism flights are now a real possibility, as are missions aimed at the major generation

of solar power in orbit for use on Earth. Also receiving serious attention is the investigation of the threat posed by Near-Earth Objects, and ways of protecting the Earth from impact by these.

2.2 The Nature of Autonomous Systems for Space Application

The foregoing brief résumé of space exploration mentions many missions, and it would probably not be an exaggeration to say that most, if not all, of these, have involved some degree of 'automation'. Even this simple comment leads immediately into difficulties - the terms automatic, robotic and autonomous can sometimes be used in an almost interchangeable way, and so it is necessary for the purposes of this thesis to state here the working definitions that will be adopted throughout this work. The three terms can be viewed as stages of increasing sophistication in systems designed to operate without real-time hands-on intervention by humans.

• **automatic** – designed to execute a predefined sequence of actions, either irrespective of circumstances, or through a predefined set of responses to foreseeable alternative conditions.

• **robotic** – able to achieve a predetermined objective within a stable environment, with minimal external intervention.

• **autonomous** – able to achieve a predetermined objective without external command or remedial intervention, and to respond to unforeseen circumstances in a way that permits continued operation in pursuit of the objective, or identification and accomplishment of new objectives, consistent with the original constraints in respect of system life and mission duration.

Additionally, it is necessary to have an understanding of what is meant by 'artificial intelligence', and the role that this plays. Unlike the other three terms, artificial intelligence should not be seen as an additional stage, but rather as a capability that permeates all three, but at increasing levels of sophistication. A definition is adopted as follows:

- **artificial intelligence** – a system competence achieved primarily through advanced programming techniques, mimicking aspects of human logic, reasoning and inference capabilities, and allowing the system to achieve enhanced levels of independent operation.

2.2.1 Characteristics of Space Applications

So what are the types of autonomy present in spacecraft, and what are the specific requirements that space applications place upon autonomous systems, and how do these influence design solutions?

Spacecraft operations encompass almost every aspect of independent operation that can be proposed. This can range from applications where pure automatic operation is acceptable, indeed desirable, to situations where the most sophisticated combinations of autonomous operation supported by artificial intelligence are applied. Rather than attempt here a comprehensive review that is not necessary for the purpose of this thesis, examples are given, within the context of the phases of a typical mission. This demonstrates the overall scope of the subject, and leads naturally into the central topic of autonomous planetary exploration operations and the role that mechanical robustness to faults can take in this.

The following sections therefore describe briefly the mission phases for a typical planetary science mission involving both planetary rendezvous and landing.

2.2.2 Autonomy and Mission Phases

The identification of mission phases is somewhat difficult, since these will vary a little depending on the mission profile. However, for a typical planetary mission with on-surface operations, it may be considered that there are seven main mission phases, namely launch, trajectory acquisition, cruise, approach and orbital injection, spin-up and eject, entry and descent, and surface deployment and operations.

2.2.2.1 Launch

This frequently imposes worst-case design environments due to the large accelerations, aerodynamic loadings and acoustic effects, that can occur. However, planetary landings may impose even higher loads on some system elements.

2.2.2.2 Trajectory acquisition

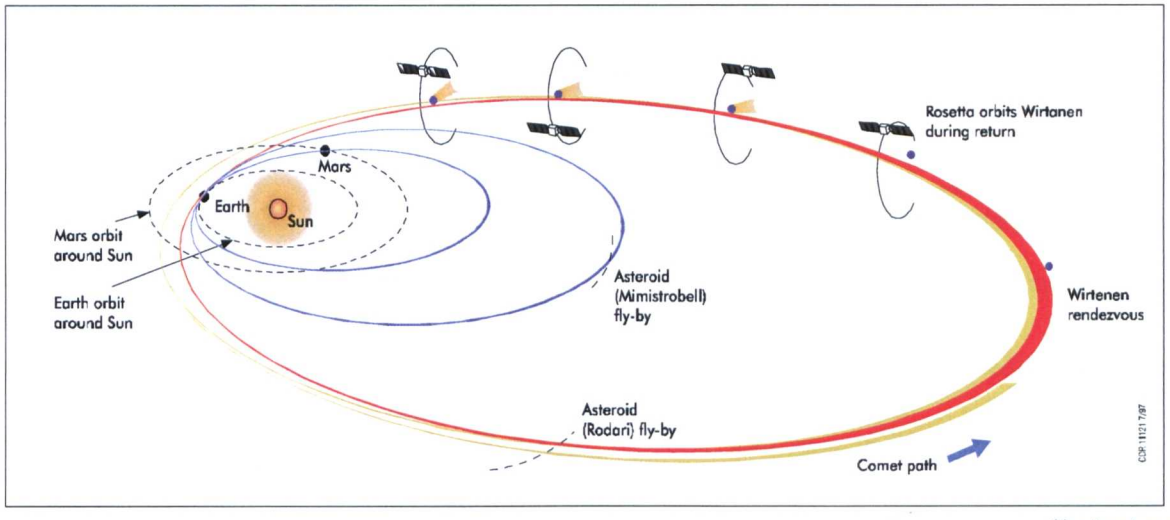
The type of trajectory acquired varies – spacecraft may be placed into long term Earth orbit, put in temporary parking orbit, or an 'orbit raising' transfer or reshaping manoeuvre undertaken. The

latter will frequently be required as a precursor to injection into cruise trajectory for planetary missions.

2.2.2.3 Cruise

The use of planetary and solar gravitational fields to accelerate spacecraft and modify their trajectory (the so-called 'slingshot' technique) has become a standard means for interplanetary missions to achieve course correction or acceleration with minimal fuel expenditure.

The method involves the spacecraft following a trajectory through the local planetary gravitational environment to provide a net increase in spacecraft kinetic energy.



(Astrium image)

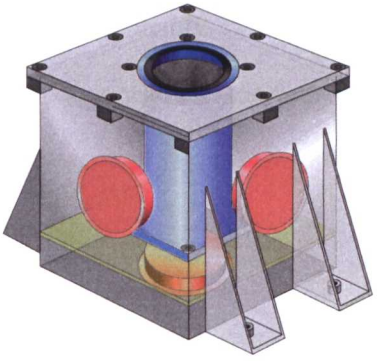
Figure 2.4: Rosetta Trajectory including Planetary Slingshots

Frequently this manoeuvre can require 'out of ecliptic plane' deflection of the trajectory that then has to be corrected after the manoeuvre, involving complex calculations and accurate on-the-spot trajectory re-calculation necessitated by any deviations from predicted planetary influences. Particular points to note are the need to accommodate very wide variation in thermal environment, carefully scheduled attitude control manoeuvres, and demanding navigational scenarios. The process is frequently repeated several times in one mission. This is primarily a **robotic operation**, although some aspects of the mission phase may need to be autonomous – as discussed in the case of autonomous star tracking, below.

2.2.2.3.1 Autonomous Star Tracking in Cruise

Most spacecraft, and particularly Earth Observation and interplanetary missions require the spacecraft to have an accurate knowledge of where it is and how it is orientated. This self-location can only realistically be done **autonomously** from on board the spacecraft.

The Autonomous Star Tracker is a high-resolution telescope with sophisticated systems for discriminating stars by measuring their brightness and spectra. The associated electronics also stores a comprehensive library of thousands of star brightnesses and their relative angular positions.



Courtesy Astrium

Figure 2.5: One Example of a Star Sensor – the Astrium Uninav System

To identify what stars the system is pointing at, the tracker carries out the following sequence:

- Sensing – The brightest star in the star tracker's field of view is identified, then the next two brightest stars in the star field are identified
- Computing - A triad (triangle of stars) is constructed, and its geometry calculated. The on-board catalogue is searched, using algorithmic search routines, for a triad that matches the one that has been identified. If necessary, a second and third triad can be identified in order to provide further orientation data. Note that parallax effects arising because the star patterns are not being viewed from Earth mean that it is necessary to have an exact point of reference if the triad relationships are to be meaningful.
- Action - The necessary corrections to the spacecraft attitude are calculated. The data is passed to the spacecraft's Attitude and Orbital Control System (AOCS). The AOCS then commands the spacecraft's angular momentum wheels and / or its thrusters to re-orientate the spacecraft.

2.2.2.4 Approach and orbital injection

Depending on whether the mission is fly-by, or is intended to enter planetary orbit, methods such as planetary 'slingshots', carried out **robotically (or autonomously?)** due to signal time delay, may apply as discussed earlier – the method can equally well be used to decelerate spacecraft so that they enter orbit, and may also be used to reorientate the orbit in situations where the desired orbit is not in the same plane as the approach trajectory. Critical control of spacecraft attitude, speed and direction are required, since there will frequently be a need either for a parent spacecraft to continue in orbit, or for it to undertake a controlled exit from the planetary locality if the mission involves travelling on to further objectives.

2.2.2.5 Spin-up and Eject

'Spin-up and Eject' are the terms applied (almost always together) to the action of the mechanisms responsible for detaching a probe from a parent spacecraft – typical for situations where the latter is intended to remain in orbit, or is continuing its journey. 'Spin-up' imparts slow rotational motion to stabilise the probe during atmospheric entry, whereas 'eject' refers to undocking the probe from the spacecraft and pushing it away. This is a critical operation where failure can lose the probe and maybe the parent spacecraft as well, especially in the worst-case scenario where they become partially detached and the spacecraft dynamics become uncontrollable or unpredictable.

The operation is achieved by relatively simple mechanisms (such as latches, springs and cams) operating automatically in concert with one another. Once initiated, this operation has to continue to completion – it involves a predetermined set of conditions – successful latch release, achievement of correct rotational speed, electrical harness detach, separation impulse spring operation, which are all required in a set sequence and with accurately determinable results and fall-back solutions. This is a good example of where **automatic operation** is a suitable solution to the problem in hand, and there is no particular need for autonomy.

2.2.2.6 Entry and Descent

Early stages of entry and descent involve a great deal of heat generation, normally combated by the use of some form of rigid (frequently ablative) heat shield. Later stages employ parachutes, air bags, and sometimes retro-thrusters. As with spin-up and eject, this stage involves a predetermined

set of actions – for example: successful heat-shield release, deployment of drogue parachute, main parachute(s) mortar firing, main parachute(s) deployment, parachute release, airbag deployment, ground impact, airbag deflation and release. These are all required in a set sequence and with accurately determinable results and fallback procedures. The scope for operator intervention, or for autonomous operation is limited, and so this too can be regarded as an example of **automatic operation** being a suitable solution to the problem in hand.

(As always, the situation is open to discussion. In the event that it is confirmed that loss of Beagle 2 was due to anomalous atmospheric conditions causing airbag failure, it can be argued that a more autonomous and less automatic sequence may have saved the mission by being able to modify airbag inflation pressures. Even so, this might still be considered to be automatic operation and not autonomous behaviour).

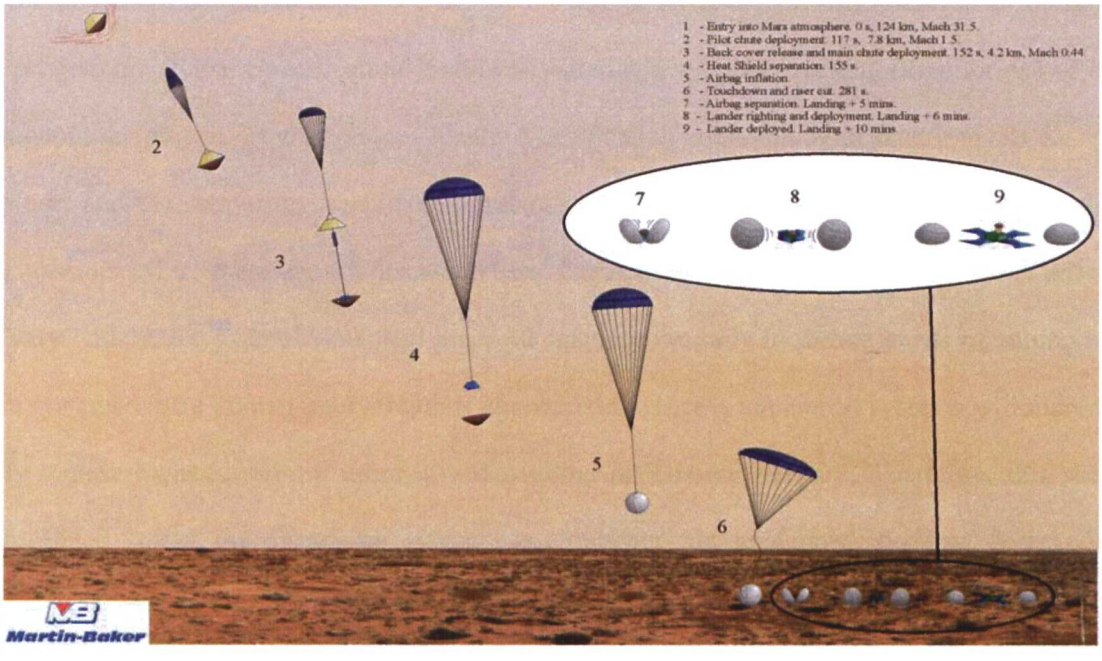


Figure 2.6: Mars Probe Entry and Descent Phases

(Martin Baker A/C image)

2.2.2.7 Surface Deployment and Operations

This is a combination of many operations, with varying degrees of **automatic, robotic or autonomous** behaviour. Such operations include the opening of access panels, deployment of solar arrays and antennae, and taking of contingency soil and atmospheric samples to allow for subsequent mission failure. The release or tidying up of landing equipment such as air bags can be a critical feature, particularly where the subsequent deployment of surface autonomous systems depends on access across such equipment.

Data reduction, encoding and transmission - Once surface operations are under way and scientific data is being generated, the lander electronics and communications systems have to undertake the ordering, coding and transmission of data back to Earth.

Command receipt - Planetary operations are highly autonomous, but occasions arise when the lander has to be commanded from Earth with concomitant difficulties arising from long time delays. The protocols for ensuring successful interaction between local autonomy and long-range mission control intervention can be complex, but also central to mission success.

The identification and resolution of design issues arising from the complex requirements associated with planetary surface autonomous operations are discussed in greater detail in the next section.

2.2.3 Requirements Identification and Tracking

Successful design definition depends crucially on careful requirements identification at the start of any programme ('Goal Specification'), and consistent and accurate tracking (monitoring) of them throughout the project. Methodologies for the identification and tracking of requirements for space missions, and the supporting verification and validation (V & V) phases are comprehensive and well documented, so these are not discussed here. Furthermore, the requirements for space missions can vary widely, for example, manned missions may require only moderate levels of autonomy in those systems where human intervention is appropriate, whereas unmanned systems will more than likely require higher levels of autonomy depending on mission purpose. Therefore, this section concentrates solely on discussing typical requirements for unmanned systems designed for planetary surface exploration, in order to provide an insight into the constraints affecting the types of system of interest in this thesis.

Although, as discussed previously, most Agencies' planetary exploration programmes include long-term plans for manned missions, the current status of planetary exploration restricts the opportunities for these. In the short-term, manned missions cannot be considered for many scenarios since a range of new technologies and sciences, such as may be required to deal with the threat to human health from extra-terrestrial organisms, need to be developed and matured in advance. Thus the role of autonomous missions remains assured for some while yet – they remain cheaper, possibly more flexible, and can be much more effective in terms of science payload

fraction delivered to the surface because none of the technological prerequisites for human life have to be carried as part of the payload.

2.2.4 General Design Objectives

Although in general terms it is not possible to define specific design objectives to be met, it is possible to identify generic requirements for surface mobility systems (eg vehicles) designed for unmanned autonomous operation on planetary surfaces. Several of these top level requirements and / or design drivers (ie requirements that dominate the design response and so 'drive' the final solution) can be determined, as identified in Tables 2.1 and 2.2. These are highly variable according to mission, and some of the tabulated data, particularly in Table 2.2 is specific to particular planets, quoted for illustration purposes only, and is not intended to be an exhaustive list.

Top Level Requirement	Comment / Typical Value
Vehicle Overall	
Target planetary body:	Moon, Inner planets – Mercury, Venus or Mars, Satellites of outer planets (eg Titan – moon of Saturn), Comets.
Mission purpose / science	Geological – multi-lander network? Exobiology.
Traverse characteristics	Range from lander and time in transit / transit profile. All highly variable according to mission type.
Mass budget	Target overall vehicle mass, and payload mass fraction – proportions of vehicle mass allocated to science payload and basic vehicle functions eg mobility, power.
Reliability targets	Operational life, reliability considerations
Operating Environment	Resistance to chemical contamination, particulate ingress / contamination, micrometeorite impact. See Tables 2.2, 2.3.
Cost and programme considerations	
Navigation and Control (including vision)	
Control philosophy	• Remote Control Systems • Menu Driven / Expert Systems • Knowledge Based/Goal Driven Systems • Evolved Autonomous Systems
External sensing	Vision, orientation etc.
Structure Sub-System	
	Payload interfaces
Thermal Control Sub-System	
Telecommunications and Data Handling Sub-System	
Transmission Time	Eg: 3 mins. at solar opposition, 22.3 mins. at solar conjunction
Power Sub-System	
Duration of surface operations	Affects basic power sub-system design – power acquisition, storage and distribution principles.
Hibernation / Standby	20 x 8 Mars Days
Power	5 to 8 Watts
Mechanisms	
Mechanisms architecture	Launcher / lander interfaces, vehicle stowage latches etc.
Lubrication	Dry lubrication may be preferable to fluid lubrication that has containment and contamination considerations.
Requirements affecting locomotion subsystem design	
Terrainability	See Table 2.3
Required surface speed	Highly variable according to mission
Mechanical architecture	Wheels, legs, tracks, etc.
Physical size and shape	Protection from transit environment, compactness

Table 2.1: A Sample of Planetary Exploration System Top Level Requirements

Table 2.1 introduced two new terms, these being 'locomotion subsystem' and 'terrainability'.

A **locomotion subsystem** is that part of a vehicle concerned with achieving its movement across a surface. Therefore it can be taken to include such items as motors, wheels, legs, suspension systems, and steering systems. In this thesis, the term 'system' is used to imply locomotion subsystem, and excludes other critical parts of a vehicle such as power management, thermal control, structure, vision and navigation equipment, computing, data links, and other 'subsystems'.

Terrainability is used as a collective term describing the ability of a vehicle to move across terrain, whether on Earth, or any other planetary body. It is used to embrace a number of different

land mobility issues, some of the key ones of which are presented in Table 2.3. Terrainability may be defined as 'a representation of the ability of a locomotion subsystem to negotiate variations in terrain, specified by the ability of the system to cater for defined categories of obstacle'. This topic is expanded upon further in Chapter 3.

Aspect	Comments
Length of Day	24.62 Earth hours
Wind Velocity	20 km/h typical 400 km/h maximum
Atmospheric Pressure	7 mb
Particulate Environment	Dust problem, meteorites less so
Temperature Variation	60 - 140 K @ poles 215 – 295 K @ equator

Table 2.2: Some Significant Features of the Martian Environment

Top level requirements of particular relevance are those affecting the design of the locomotion subsystem, and these are expanded upon in Table 2.3:

Terrainability Consideration	Typical Specified Limits	Subsidiary Parameters / Design Drivers
Longitudinal slope traverse - uphill	15°	• Power draw • Stability / self righting
Longitudinal slope traverse - downhill	20°	• Stability / self righting
Lateral slope traverse	15°	• Stability / self righting
Mobility in deep dust pockets	Variable depending on nature of landing site. Locomotion with 50% of wheel submerged may be typical.	• Power draw • Contamination - not amenable to analysis, more the subject of good design practice • Wheel sinkage • Stability / self righting
Step Obstacle negotiation/ Trench climb-out	100mm height	• Power draw • Stability / self righting
Hole Spanning	100mm span	• Locomotion s/s geometry • Stability / self righting

Table 2.3: Typical Locomotion Sub-System Terrainability Requirements

2.3 A Simple Overview of Control System Classification

In selecting design methodologies for autonomous robotic systems, a choice of control strategies presents itself. These must be selected according to the nature of the mission, its objectives, and the degree of risk considered acceptable in achieving those objectives. Also having a direct bearing on the choice of **control strategy** is its match to the **proposed kinematic architecture**, and the advantages which each can bring to the other in terms of improved flexibility, reliability and robustness, increased development promise, cost effectiveness, etc.

Typically, the navigation problem proves central to the choice of control system, and this underlies the approach to categorisation put forward here. However, other system functions such as power system management, thermal control, and payload management, all exert their influences on control system design. Available strategies can be loosely categorised as follows, although in practice, elements of all approaches will be intermingled as appropriate to the task in hand:

- **Remote Control Systems**
  Here, control of the remote system is achieved using direct command by the ground segment (that is to say, all the Earth-based infrastructure, including remote command and control systems), with no local functionality. Although simple, such systems have a very strong heritage, and, where signal delay times are reasonable, constitute very strong contenders for selection. The direct command mode can be operated in conjunction with ground segment or planet segment generated Digital Terrain Modelling techniques, which renders the system extremely effective in some circumstances.

- **Menu Driven / Expert Systems**
  With this strategy, the system exhibits limited local autonomy, based on reference to pre-defined look up tables and diagnostic routines. However, the system's autonomy is limited entirely to catering for foreseen and strictly defined situations. This approach may be more strictly defined as 'automatic' – in reality, there are no clear divisions, and control system design features specific to particular system designs may sway the argument one way or the other.

- **Knowledge Based / Goal Driven Systems**
  Such systems exhibit full local tactical functionality independent of ground segment intervention. The system is capable of on-board solution of problems based on prior ground segment definition of strategy in goal orientated terms.

- **Evolved Autonomous Systems**
  A more advanced, and potentially very powerful, approach to goal orientated autonomy is offered by current research epitomised by that at the University of Sussex, Falmer, UK, into the development of control systems using evolutionary software approaches. Here, control software is developed through progressive generations using quasi-genetic techniques. By selection of those solutions most fit in achieving the defined system goals, and further evolving

these strains, control system behaviours optimised towards the chosen goals can be obtained. This work was also referred to in Section 1.3.

2.4 Mechanical Architectures

Clearly, control methodologies cannot be satisfactorily selected without reference to the intended mechanical design of the projected system. Matching of the control system to the required task, and to the vehicle design selected to execute it, is central to successful mission completion.

Classification of the many design concepts developed for a planetary vehicle is problematic. For the initial purposes of this discussion, a mechanically based approach, identifying ten different 'terrainability system architecture' categories is adopted (see table below). The attributes of each of the approaches are discussed in the following paragraphs.

Classification	Comment / Description
Type 1(n)	Simple chassis, with n wheels
Type 2(n)	Advanced chassis, rocker bogie, with n wheels
Type 3(n)	Advanced chassis, fork wheel, with n wheels
Type 4(n)	Advanced chassis, articulated and / or jointed, with or without extension capability, with n wheels. May include elastication and multiple units.
Type 5	Tracked concepts
Type 6(n)	Walking Machines - biological (stick insect) type, n legs
Type 7(n)	Walking Machines - non-biological type, n legs
Type 8	Crawlers
Type 9	Hybrids
Type 10	Miscellaneous

Table 2.4: Classification of Planetary Exploration Vehicles by Mechanical Morphology

2.4.1 Wheeled Vehicles

Wheeled vehicles are considered to comprise two groupings - those with essentially simple chassis, and those where more advanced chassis designs have been incorporated. The division between these two tends to be very blurred, but the intention here is to discriminate between essentially rigid chassis structures (allowing, of course, for normal engineering strain and deflection considerations), and those designed to have variable geometry.

2.4.1.1 Simple Chassis

Although even very early studies rapidly realised the advantages of advanced chassis designs, these were not immediately implemented, mainly for the standard reasons of cost, programme feasibility,

and timescale. Hence, early rover concepts, notably the USA and USSR lunar applications of the '70's, made use of relatively straightforward chassis designs with largely fixed mechanical geometries. This type of vehicle has the disadvantage that it requires landing sites chosen as suitable for operation of the intended vehicle. Alternatively, sites can be used which have undergone prior preparation. Although missions involving such preparation have been discussed, this has not yet been undertaken. If this is not to be the case, then either a driver must be supplied (Apollo Lunar Rover), or else the vehicle must be designed for remote operation (as defined previously) with a human in the loop (Lunakhod). Because of transmission time delays, this is only realistic for exploration of very close bodies, ie the Moon, where the 3-second delay may be acceptable, depending on the system design and control strategy adopted. Once exploration of more remote bodies is considered, then more advanced control approaches, such as those already discussed, need to be adopted. Alternatively, vehicle speeds have to be reduced substantially to allow for adequate ground-segment response times to unforeseen events.

2.4.1.2 Advanced Chassis

With the very high mission costs involved, maximising scientific return by targeting missions to more varied geological regions is an obvious advantage. This involves driving under much less benign conditions. Furthermore, local rover autonomy is bound to introduce uncertainties about the exact nature of the terrain being encountered. This has led to a large number of more advanced chassis designs. Extra chassis complexity can allow greater freedom of movement under autonomous control, especially when combined with powerful local computing linked to vision and navigation systems. This approach is typified by the 'rocker bogie' design of JPL's 'Rocky' series, especially, most recently, the Mars Pathfinder 'Sojourner', and 'Spirit' and 'Opportunity' vehicles. Here, improved mobility is achieved by advanced springless suspension techniques that introduce a large degree of variable chassis geometry.

Even greater flexibility is achieved with designs such as the Mars '98 'Marsokhod' design (see Appendix H) tested in the Mojave Desert. This employs a three segment articulated chassis, plus extensible wheelbase. This latter feature introduces a limited crawling ability in the event of loss of wheel traction. Variable centre of gravity control can also be used to improve the versatility of a design, since this considerably increases the size of obstacle that can be successfully negotiated.

This is the approach adopted by JPL's Go-For design, which employs a "fork wheel" system allowing the wheel positions relative to the vehicle centre of gravity to be changed. This will produce the same effect as an extensible wheelbase, but via different mechanical means.

2.4.2 Tracked Vehicles

One of the first design concepts to apply tracked vehicle principles to planetary exploration is that postulated by Northrop in the USA as part of NASA's Mobile Lunar Laboratory (MOLAB) programme in the late 1960's. Prior to this, research into track / surface interactions had been carried out as early as 1961 at General Motors' (USA) Land Mobility Laboratory by Becker [8]. Even earlier work was conducted, but tends to be irretrievably entangled with strictly Earth-orientated projects, much of it with military objectives. Although one would suppose that tracked vehicles offer a viable solution to the problems of driving over planetary or cometary terrains, no current concepts have been identified at this time. Although the Instrument Deployment Device (IDD) for ESA's Marsnet / Rosetta programme [15, 16] appears at first sight to be tracked, this is not the case, and the vehicle is more correctly classified as a hybrid. (Note that the names IDD and Nanokhod were both applied to this vehicle, which is that referred to in Appendix H). Given the number of design teams involved in planetary vehicle development, it must be considered that the scarcity of tracked designs indicates that this approach is fundamentally flawed for this type of application, but major drawbacks are not immediately obvious.

The approach offers several potential advantages such as low centre of gravity leading to increased stability in low gravities, although it can be argued that other designs can achieve equivalent stability. Robustness of tracked systems is generally good, and terrainability is also a strong point. On the negative side, it may be that tracked vehicles tend to carry an extra weight penalty, and it is unarguably the case that the design of track systems for operation in very dusty environments presents a number of demanding technical challenges, and manoeuvrability may be limited in some circumstances. However, tanks have traditionally played a major role in desert warfare, so these problems are clearly not insurmountable in the appropriate circumstances.

2.4.3 Walking Vehicles

Walking machines of one form or another have been around for a long time, but only comparatively recently with the advent of sophisticated control systems have they begun to show real promise. Two versions may be said to exist - those that emulate systems in the natural world, particularly insects, in their articulation, and those which do not. The former type seems to hold out most promise, although they suffer from considerable mechanical complexity, and also present a number of problems from the mathematical modelling viewpoint. It is in this area that major advantages may be seen to exist for the application of advanced autonomous control systems.

2.4.4 Crawling Vehicles

A potential category of vehicle that seems at the present to have few representatives is the crawler. This term signifies machines which do not employ wheels, tracks or legs, but move by other means such as varying the body geometry in conjunction with establishing a progression of holds, anchor points, or simply frictional reaction on the planetary surface. Since this method requires constantly establishing reaction points against which the vehicle can lever itself, it may be argued that the vehicle has a more positive attachment, or at least a closer proximity to the surface than other techniques. This may offer significant advantages in low gravity environments.

2.4.5 Hybrid Vehicles

Many options present themselves for achieving motion by combinations of the previous categories. Devices in this category achieve motion by techniques such as peristalsis, variable wheel geometry, moveable spines etc. There is a real possibility that a number of potentially useful designs could lie in this area, but that they might not be developed because they seem to be too fanciful. Care must be taken to achieve the correct balance between brainstorming and flights of fancy.

The most significant representative of this classification is currently the Transmash design for the Marsnet / Rosetta IDD [17, 18]. Another example of a hybrid vehicle is the "Hopper" design adopted for the Russian Phobos project. Unfortunately, with the loss of the mission, these devices never saw 'active service'.

Chapter 3
Kinematic Aspects of Space Systems

Chapter 2 provided a general review of space systems history and progressed from this to examine some of the planetary exploration issues that can arise, treating these from a top-level, requirements-based viewpoint. The concepts of 'locomotion subsystem' and 'terrainability' were briefly introduced, and it was discussed how the kinematics[1] of a vehicle's 'locomotion subsystem' is central to its ability to cope with the terrainability requirements that a planetary surface exploration mission will need to satisfy. To allow for a more structured analysis of the various mechanical kinematic system topologies that planetary exploration vehicles employ as locomotion subsystems, and in particular, how such systems behave under fault conditions, it is necessary to establish what mathematical 'tools' can be brought to bear on this issue.

This chapter examines in greater detail some of the more significant aspects of locomotion subsystem kinematic design, and continues by identifying some of the 'tools' available, and identifying potential analysis methods. In particular, it will be shown how a significant proportion of graph theory is relevant to this problem, and, when supplemented by aspects of linear algebra, offers powerful methods for application to this task. It will also be shown that developments of graph theory can be established, capable of providing an enriched model of a system's kinematic behaviour.

3.1.1 Terrainability of Space Locomotion Systems

In Chapter 2, Sections 2.2.3 and 2.4, a general overview of terrainability requirements and some available mechanical kinematic architectures for responding to planetary mission requirements were described. Of necessity, only a limited selection of vehicle types was covered - a full

[1] Kinematics – concerned with the analysis and synthesis of motion.

treatment of all possible designs is not necessary to establish the general point regarding the scope of design that these systems cover, and in any case, would occupy the entire thesis if treated fully.

The major terrainability requirements or categories applicable to planetary exploration vehicle locomotion subsystems, in practice almost always occur in combination. However, some of the key elements, touched upon in Chapter 2, can be separated out and identified as being:

- Longitudinal slope traverse, uphill / downhill - power draw
- Longitudinal slope traverse, uphill / downhill – stability / self righting
- Lateral slope traverse – stability / self righting
- Mobility in deep dust pockets – power draw, contamination, wheel sinkage, stability, self righting
- Step obstacle negotiation – power draw, stability, self-righting
- Trench climb-out – power draw , stability, self righting
- Hole spanning

These categories are largely self-explanatory, but are illustrated for clarity, using a wheeled robot as an example, in Figure 3.1. Analysis methods for many aspects of these terrainability requirements are available based on classical mechanics[1], and these can be utilised quite effectively. However, these are not expanded upon here since they are well documented, and not of direct relevance to the arguments being advanced in this thesis.

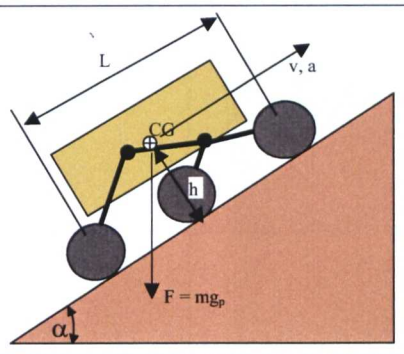
The basis of the argument here is that, whilst classical mechanics treatments, and many other advanced methodologies are available or are in course of development, they do not address certain fundamental issues associated with system organisation, and scope exists for development of additional treatments based on morphology[2] and topology[3], for the reasons discussed later.

---

[1] Mechanics – Concerned with the motion and the causes of motion of physical objects. (Dynamics – concerned with the physical causes of motion. Statics – concerned with the conditions under which no motion occurs).

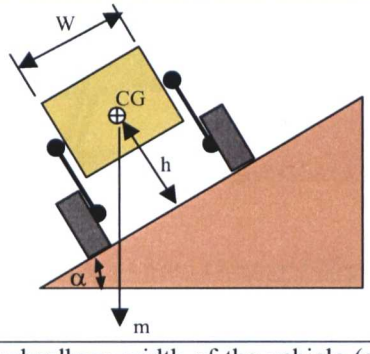
[2] Morphology – the study of physical or mechanical structure and form.

[3] Topology – here used to refer to the study of those properties relating to the mechanical interconnections of systems.



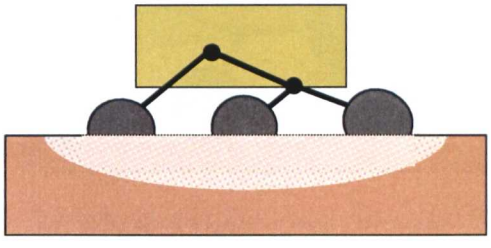
m = mass of vehicle
g_p = acceleration due to gravity on planet
L = wheelbase of vehicle (assumed symmetric)
h = normal distance of cg to the ground line
v, a are velocity & acceleration up the slope
α is the angle of the slope

(a) Longitudinal Slope Power Draw and Stability



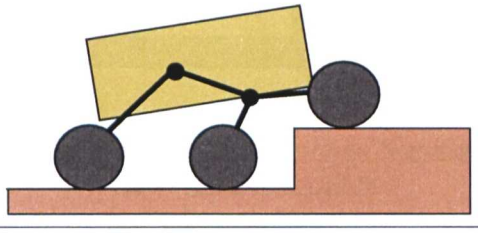
W is the wheelbase width of the vehicle (assumed symmetric)

(b) Lateral Slope Stability

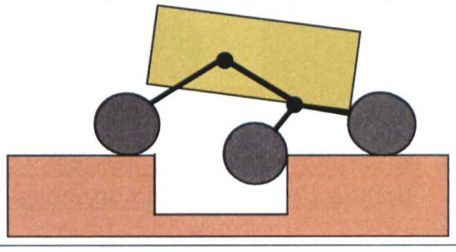


The subject of dust pocket mobility is highly complex and covered by a number of references, in particular Ageiken [1], Bekker [8], and Berkelman et al [10]

(c) Dust Pocket Mobility



(d) Step Obstacle Negotiation

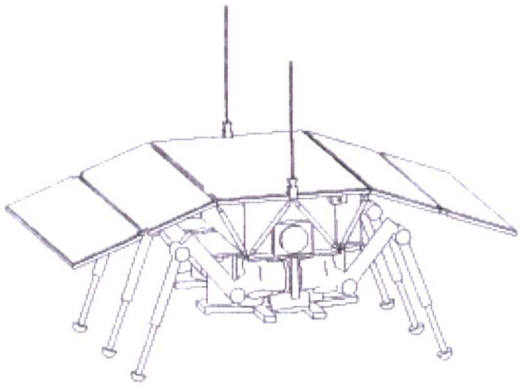


(e) Hole Spanning

Figure 3.1: Overview of Terrainability/Mobility Issues for a Typical Wheeled Robot

It can be seen that each of these terrainability categories raises issues corresponding to several of the terrainability definitions given at the beginning of this chapter. Taking one of these, 'step obstacle negotiation', for which the wheeled vehicle nominal condition is presented in Figure 3.1(d), and examining it in greater detail from the point of view of both legged and wheeled systems, it is possible to obtain a better understanding of the issues involved. In Figure 3.2, the

hexapod 'Millennium Hero' design study and the wheeled 'Generic Rocker Bogie' designs are used as examples, and their behaviour under just one simple fault condition is examined.



(a) Legged Robot – Millennium Hero

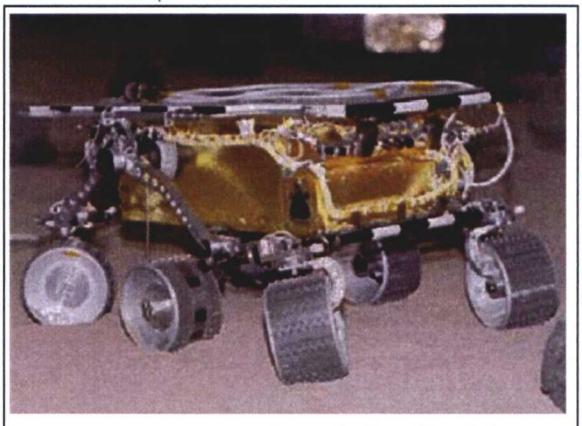
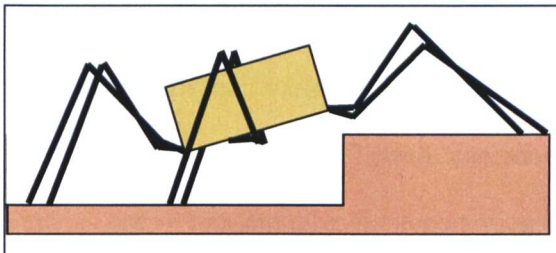
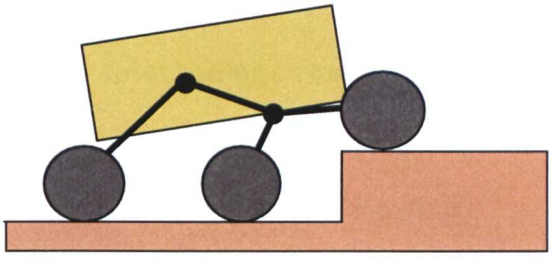


Image: Jet Propulsion Laboratory

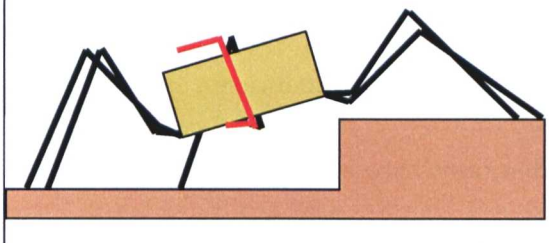
(b) Wheeled Robot – Generic Rocker Bogie



(c) Legged Robot - Step Obstacle Negotiation

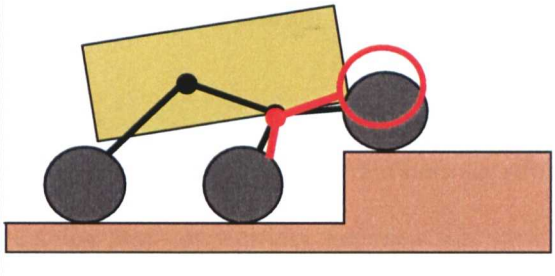


(d) Wheeled Robot - Step Obstacle Negotiation



(e) Legged Robot - Step Obstacle Negotiation under Fault Condition

(One Central Leg Failed)



(f) Wheeled Robot - Step Obstacle Negotiation under Fault Condition

(Wheel lost from one bogie)

Figure 3.2: Examples of Legged and Wheeled Robot Terrainability Considerations under a Fault Condition

Consideration of the states illustrated shows that qualities of the two systems contributing to their ability to negotiate obstacles can conveniently be examined under the categories of geometry[1],

[1] Geometry – the branch of mathematics concerned with the properties of space, and of figures in space.

dynamics, kinematics, morphology, topology, terrainability and gait[1]. The various qualities described by these combine to define the systems' locomotion capabilities as follows:

- The **geometry** of the two systems, together with their scale relative to the obstacle dictates whether the systems have the 'reach' necessary to scale or span objects

- The **dynamics** of the systems together with the available power, dictates whether the necessary energy is available to make the positional and velocity changes necessary to span the obstacle. The power train and power transfer systems determine whether the system can apply this power in a way that facilitates the manoeuvres necessary - for example, the ability to create the necessary frictional forces through ground interfaces.

- The **kinematics** of the systems describes their motion in terms of the positions, velocities and accelerations of their various components. This may have a direct bearing on terrainability, since it is not solely a static issue. Terrain can move – slip, slide or subside - under the presence of a vehicle, and in this situation, the locomotion system's kinematic capabilities can be critical.

- System **morphology** – directly affects the way in which a kinematic system is able to 'approach' the task, and controls issues such as whether the system has adequate articulation, and whether enough legs / wheels are available to maintain the system's pose and posture / stance whilst other appendages are repositioned.

- System **topology** – dictates what options the systems have for dealing with terrain because of the way that it describes and controls the inter-relationships and connectivity between the various appendages, what mobility they have, the effects of any faults, and the form of movement that the system is capable of effecting. In the fault cases illustrated later in this thesis (eg Chapter 5 and Chapter 8), topology affects how fit the systems are for operation in fault modes. It will decide whether the systems have the redundancy or alternative modes of mechanical operation that will allow their control systems to move into an alternative mode of locomotion, by altering gait, for example, or by modifying its internal system model to allow for the fact that system stability characteristics will have changed due to loss of, say, the leg or wheel.

---

[1] Gait – the leg planting pattern adopted by animals and legged robots in order to locomote across terrain and to perform manoeuvres.

3.1.2 The Relevance of Gait to Terrainability

To develop the previous discussion further, it can be seen that, in the case of a legged system, its ability to locomote across the required terrain is strongly dependent on its gait – the ability to change gait is a fundamental enabler of arthropod fault (ie injury) operational modes. Frequently, the structures of robotic legs are based on those of insects, and as a direct result of this, robotic gaits tend to be modelled on those of insects also. Insect legs take the general form shown in Figure 3.3, and frequently, legged robots utilise very similar structures, as also illustrated in Figure 3.3 by the inclusion of an illustration of a design study for a leg mechanism for use with the Millennium Hero hexapod concept (see Appendix H).

Many walking robots with non-biological structures have also been developed, however, the intention here is not to give a comprehensive résumé of walking robots, but simply to illustrate the general relevance of kinematic structure to robotic motion behaviour.

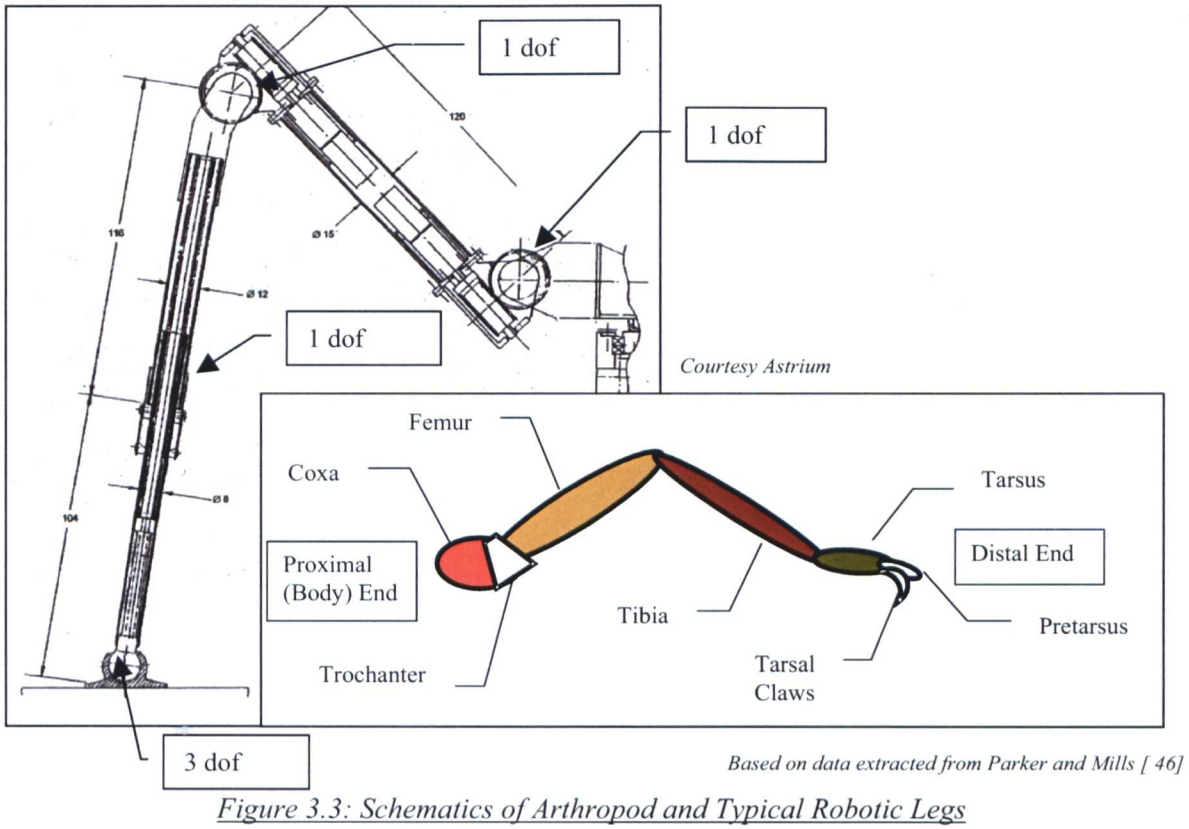


Figure 3.3: Schematics of Arthropod and Typical Robotic Legs

(Four main articulated segments of arthropod leg shown in different colours)

Figure 3.4 illustrates two gaits typical of terrestrial arthropods, which are frequently adopted for use in legged robots with six legs or more. The figure shows the metachronal and tripod gaits as applied to a hexapod robot [46]. The tripod gait is normally quoted as that which provides

maximum stability on the ground, and the metachronal gait is that which is most clearly demonstrated by video of centipedes and millipedes in action, where the legs on each side of the body exhibit a wave motion starting at the rear of the body and moving forwards. These are only two sample gaits – the study of locomotion by living creatures, and the emulation of such creatures by robotic systems, is an extensively studied field - the literature is very comprehensive, and there is no attempt made here to summarise it all, since examples are sufficient for the purposes of this thesis.

In Figure 3.4, a topological diagram of an arthropod (or, equally well, a legged robot) is shown, with legs represented as simple, single jointed, mechanisms - no attempt is made to accurately represent the leg linkages. Interactions between the ground and the system are shown as additional lines in the diagram in order to illustrate the leg planting pattern of the tripod and metachronal gaits. These interactions, as in the case of wheeled vehicles, are complex and depend on issues including soil compaction and cohesion, foot coefficients of friction, leg angle, and power transfer requirements.

The leg identification system used in Figure 3.4 is shown in Figure 3.4(a):

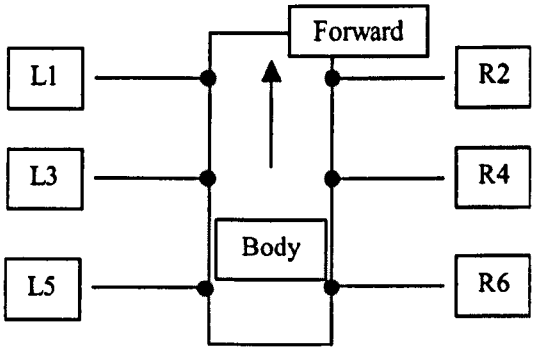


Figure 3.4a: Key of Leg Identification Labels for Figure 3.4 (L = left, R=right)

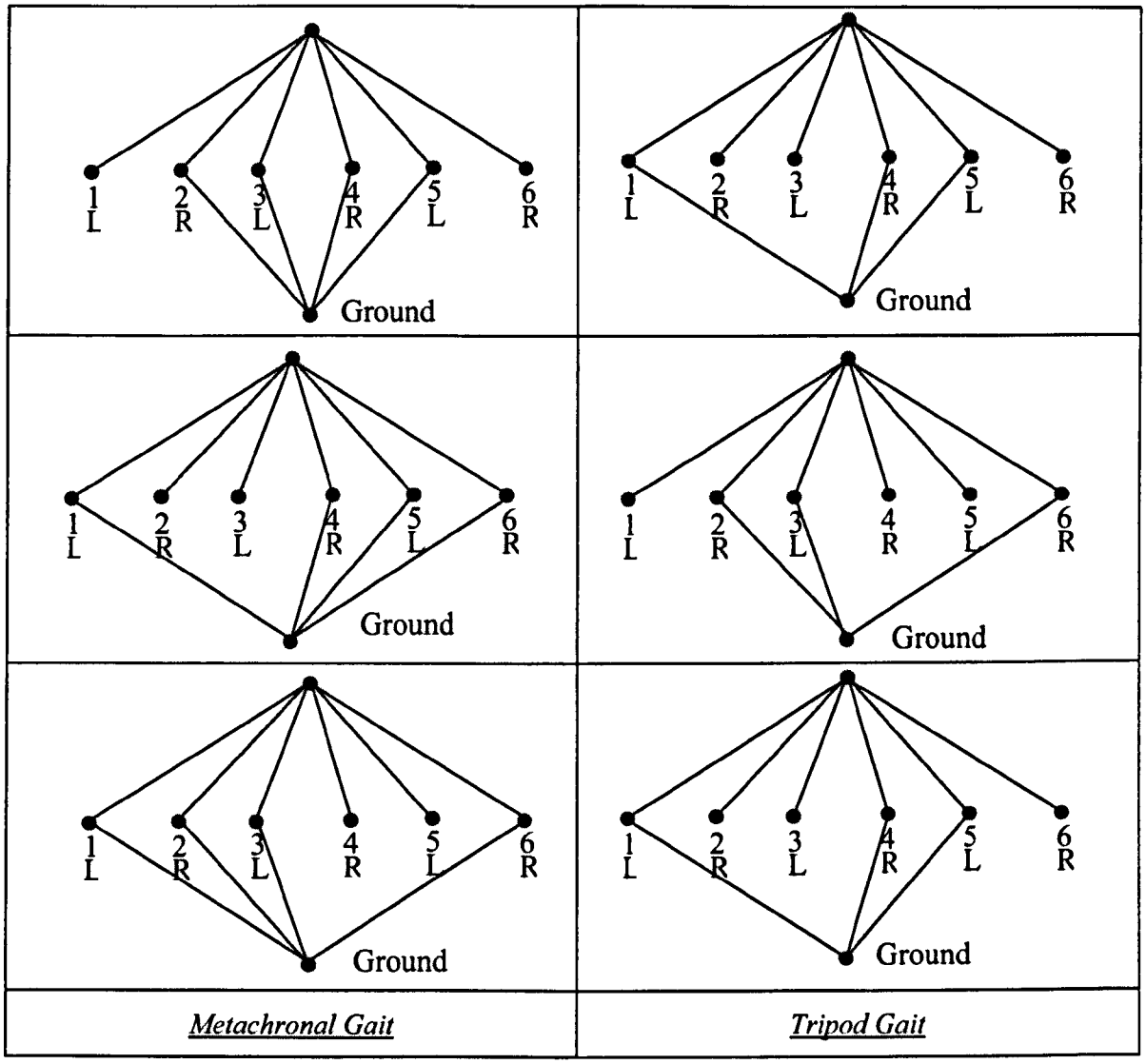


Figure 3.4: Topological Diagram of the two most common Gaits for Terrestrial Arthropods

Legged robots also illustrate well the interaction between control systems and topology. In order to execute the various required gaits, sophisticated control systems, for example neural networks, are frequently employed, and for these control technologies to operate successfully requires corresponding insight into the underlying structure and connectedness of the system, for which a topological approach is well suited.

The diagrams shown above, only illustrate the 'nominal operating mode' of the insect or robot. The gaits shown, in terms of leg planting and ground interaction sequence will be modified by any injuries (in the biological case) or faults (in the mechanical case).

The kinematic topology of the fault modes of all forms of robotic locomotion systems is an important topic which will be discussed further in Chapter 5.

3.2 Basic Graph Theoretical Concepts

This chapter now identifies a number of graph theoretical methodologies suitable for application to the topological analysis of kinematic systems. Some general observations are made, and working definitions of some of the Graph Theory terms used are provided, before proceeding. Relevant supporting references are provided at appropriate points throughout the text.

3.2.1 Kinematic Chains, Direct Graphs and Interchange Graphs

The fundamental step in applying graph theory to the kinematic chain of a locomotion subsystem is the generation of its interchange graph representation, where the rigid bodies (links) and joints of the chain are represented by graph vertices and edges respectively, [14]. A simple example of this is provided in Section 3.4.1.

It should be noted that in this thesis, the interchange graphs of the various systems discussed are shown for clarity without representation of the ground links. Consequently, the joint created by the interface between a leg or wheel, etc, and the ground, as illustrated earlier in Figure 3.4, is not shown. However, it should be recognised that complete analysis would require this, plus representation of ground movement due to stone rolling, soil compaction, and similar effects.

3.2.2 Kinematic Mobility

3.2.2.1 Reuleaux Joints (Kinematic Pairs)

Franz Reuleaux (1829 – 1905) defined six types of surface contact joints between rigid bodies, the so-called 'lower kinematic pairs' [54], and these form a fundamental building block for much of the work in this thesis. The reference contains a detailed description of these, and it is not considered necessary here to repeat this information, but the type names, degrees of freedom, and methods of reference in this work, are summarised in Table 3.1:

Reuleaux Joint Type (J)	Number of Degrees of Freedom	Joint Symbol	Drive Notation	
			Driven joint Drive on	Driven joint Drive off
Revolute	1	R	R $d_{(on)}$	R $d_{(off)}$
Prismatic	1	P	P $d_{(on)}$	P $d_{(off)}$
Screw (helical)	1	H	H $d_{(on)}$	H $d_{(off)}$
Cylindric	2	C	C $d_{(on)}$	C $d_{(off)}$
Spherical	3	S	S $d_{(on)}$	S $d_{(off)}$
Planar	3	E	E $d_{(on)}$	E $d_{(off)}$

Table 3.1:The Six Reuleaux 'Lower Kinematic Pairs', and Joint Notation Adopted

3.2.2.2 Mobility Definitions

The concept of kinematic mobility [54], which represents the degree of movement available to any specific kinematic system is of direct relevance to the analysis of planetary locomotion systems. A commonly used measure of the mobility of a three-dimensional kinematic, system is given by the Kutzbach criterion:

$$M = 6(n-1) - 5j_R - 5j_P - 5j_H - 4j_C - 3j_S - 3j_E \quad (3.1)$$

Where n is the total number of rigid bodies, and j_R, j_P, j_H, j_C, j_S, and j_E are the total numbers of revolute, prismatic, screw (helical), cylindric, spherical, and planar joints respectively.

In the case of a two-dimensional system, only revolute and prismatic joints are considered, and Equation 3.1 is replaced by the Grübler criterion:

$$M = 3(n-1) - 2j_R - 2j_P \quad (3.2)$$

where the symbols have the same meaning as in Equation 3.1. The term 6(n-1) in Equation 3.1 reduces to 3(n-1) in Equation 3.2 because only three degrees of freedom are available to each link in a 2-dimensional system, and not six degrees of freedom as would be the case in 3D.

3.2.3 Spanning Trees and Spanning Forests

In graph theory terminology, a forest is defined as a disconnected graph that does not contain any cycles. A connected forest is termed a tree. The spanning tree of a graph is the minimum subgraph connecting all the vertices of the graph, such that no further edges can be removed without the

graph becoming disconnected. By the same token, a spanning forest is a forest in which removal of any edge will result in the forest becoming further disconnected [88].

3.2.4 Fundamental Cycles and Fundamental Cutsets

The set of fundamental cycles of a graph, G, relevant to any one of that graph's spanning trees, T, is the set of cycles produced by adding in turn each edge of G that was excluded in forming T. A cutset of G is a set of edges whose removal from G increases the number of components of G, ie if G is initially connected, then two components are produced, and if G is already disconnected, the number of components is increased by one.

If the graph G is associated with a spanning tree, T, the removal of any edge of T divides the vertex set of T into two disjoint sets, V1 and V2. The set of all edges of G joining a vertex in V1 to a vertex in V2 is a cutset of G, and the set of all cutsets obtained in this way, by removing separately each edge of T is the fundamental set of cutsets associated with T [88].

$$\text{Number of fundamental cycles} = m - n + 1$$

$$\text{Number of fundamental cutsets} = n - 1$$

(Where n is the number of vertices in the original graph, and m is the number of edges in the original graph).

3.2.5 Adjacency Matrices and Incidence Matrices

The adjacency matrix is defined as the n x n matrix whose ij-th term is the number of edges directly joining vertex i and vertex j, [88]. Similarly, the incidence matrix is defined as the n x m matrix whose ij-th term is 1 if vertex i is incident to edge j, and 0 otherwise. Note that the numbering of the interchange graph vertices and edges does not matter, since equivalent matrices will result from any consistent approach. Equivalence can be demonstrated by showing that any adjacency matrix of a system can be derived from any other for the same system by any permutation of rows followed by the same permutation of columns, [90]. In this thesis, the concept of incidence matrices is not utilised.

In the type of system being considered here, the terms of the various adjacency matrices are always unity, since this is typical of the types of connection being considered. Nonetheless, it ought to be

borne in mind that based on the above definition, this need not be the case, and this is particularly relevant when considering the concept of the Constraints Matrix, **C**, which will be introduced in Section 3.4.3.1.

3.2.6 Degree Sequence

The degree sequence of a graph is the sequence generated by sequential enumeration of the degrees (ie number of connections) of the vertices of the interchange graph of a kinematic system, [88]. Conventionally one commences with the lowest degree, and works up.

3.2.7 Isomorphic and Cospectral Systems

Where the graph connectivity of two systems is identical, then they can be regarded as isomorphic [71]. If two systems have the same characteristic polynomial (defined in Section 3.3.1), then they are cospectral. Cospectral systems are not necessarily isomorphic [21].

3.3 Some Relevant Aspects of Linear Algebra

In order to be able to take full advantage of the graph theoretical concepts described in the previous section, certain features of linear algebra are also called upon, as follows:

3.3.1 Characteristic Polynomials and their Coefficients

The concept of characteristic polynomials and their coefficients is very powerful, and is used as a fundamental building block for the work discussed in this paper.

Yan and Hall [90] define the characteristic polynomial, $P(\lambda)$, of an adjacency matrix, **A**, as the determinant of the characteristic matrix, CM, where $CM = \lambda\mathbf{I} - \mathbf{A}$, so that:

$$P(\lambda) = |\lambda\mathbf{I} - \mathbf{A}| \quad \text{.....(3.3)}$$

where λ is a dummy variable, and **I** is a unit matrix of the same order as **A**. It can be shown that the coefficients of the characteristic polynomial have specific physical meanings, and can be evaluated by the "Coefficients by Inspection" approach suggested by Yan and Hall, as follows:

- Leading coefficient, p_o, is always unity
- Second coefficient, p_1, is always zero

- Third coefficient, p_2, is the negative of the number of connected pairs in the chain.
- Fourth coefficient, p_3, is the negative of twice the number of three pair loops.
- Fifth coefficient, p_4, is the number of two separated connected-pairs, minus twice the number of four pair loops.
- Sixth coefficient, p_5, is the negative of twice the number of five-pair loops, plus twice the number of groups formed by one three-pair loop and one connected pair.
- (Higher coefficients also have physical meaning [90])

In practice, generation of characteristic polynomials by inspection is not a realistic option for systems with more than just a few vertices, and proprietary software packages such as Maple or Mathcad provide a more suitable means of extracting the required expressions.

3.3.2 Eigenvalues and Eigenvectors

Any characteristic polynomial, $P(\lambda)$, has its associated characteristic roots, eigenvalues, and eigenvectors. These are not graph theoretical concepts, but rather an established feature of linear algebra. It will be seen later in this thesis, that the concepts of both eigenvalues and eigenvectors have an important role to play in the characterisation of locomotion systems.

Eigenvalues and eigenvectors are centrally involved in linear mappings between spaces, as outlined below.

Consider any 2D space with the special points P and Q defined by orthogonal position vectors

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Taking the product of each of these vectors and the matrix $\begin{bmatrix} a & c \\ b & d \end{bmatrix}$ gives:

for point P: $\begin{bmatrix} a & c \\ b & d \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} a \\ b \end{bmatrix}$ and for point Q: $\begin{bmatrix} a & c \\ b & d \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} c \\ d \end{bmatrix}$

Thus, the original orthogonal vectors are transformed into the columns of the matrix (Figure 3.3).

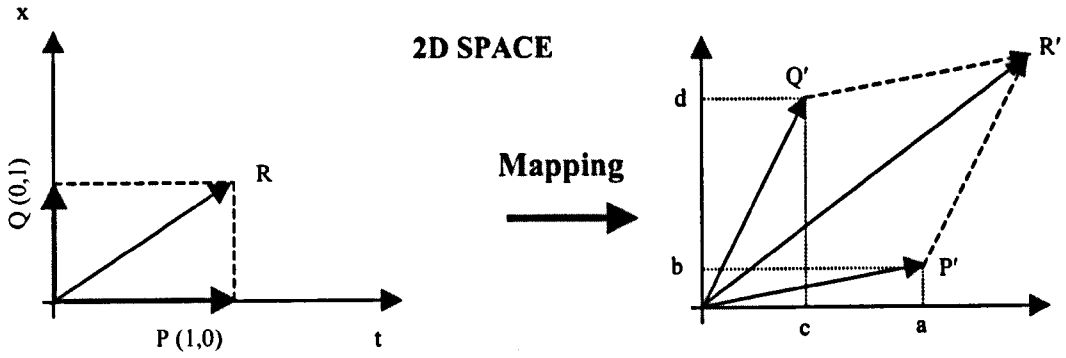


Figure 3.3:Points Mapped by a Matrix to Other Points

Now let the characteristic polynomial of the matrix $\begin{bmatrix} a & c \\ b & d \end{bmatrix}$

have eigenvectors $\underline{e} = \begin{bmatrix} e_1 \\ e_2 \end{bmatrix}$ and $\underline{f} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$

so that the matrix of eigenvectors is $\begin{bmatrix} e_1 & f_1 \\ e_2 & f_2 \end{bmatrix}$

$$\begin{bmatrix} a & c \\ b & d \end{bmatrix}\begin{bmatrix} e_1 \\ e_2 \end{bmatrix} \rightarrow \begin{bmatrix} \lambda_1 e_1 \\ \lambda_1 e_2 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} a & c \\ b & d \end{bmatrix}\begin{bmatrix} f_1 \\ f_2 \end{bmatrix} \rightarrow \begin{bmatrix} \lambda_2 f_1 \\ \lambda_2 f_2 \end{bmatrix}$$

So that the points $\begin{bmatrix} e_1 \\ e_2 \end{bmatrix}$ and $\begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$ are mapped to $\begin{bmatrix} \lambda_1 e_1 \\ \lambda_1 e_2 \end{bmatrix}$ and $\begin{bmatrix} \lambda_2 f_1 \\ \lambda_2 f_2 \end{bmatrix}$ by the matrix $\begin{bmatrix} a & c \\ b & d \end{bmatrix}$

where λ_1 and λ_2 are the eigenvalues of $\begin{vmatrix} a & c \\ b & d \end{vmatrix}$, [3]

3.4 Representation of Kinematic Geometry using Graph Theory

3.4.1 Graph Theoretical Description of Undriven Dyad

Applying the graph theoretical concepts described in 3.2, it is now possible to create a basic kinematic description for a system. Consider the simple example of a dyad – that is to say a minimal system comprising one joint, in this case revolute, and two bodies joined at the revolute, as shown in Figure 3.4.

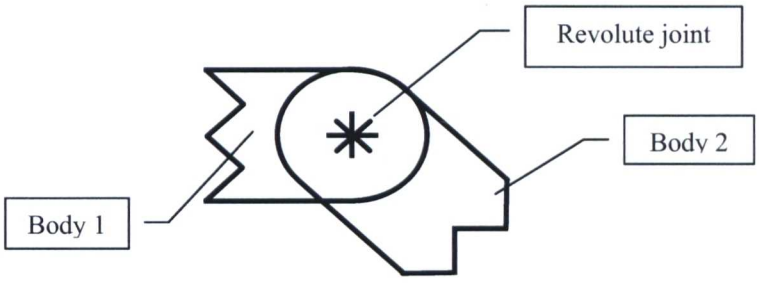


Figure 3.4:Diagram of Undriven Dyad

The interchange graph is formed by vertices representing bodies, and edges representing joints:

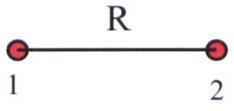


Figure 3.5: Interchange Graph of Undriven Dyad

The degree sequence for the interchange graph of an undriven dyad is simply (1,1).

Applying the standard mobility formulae (Equations 3.1 and 3.2) to the undriven dyad, it can be seen that the 2D mobility is $3(2-1) - 2*1 = 1$.

The adjacency matrix, A, for the undriven dyad is as follows:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \text{.....(3.4)}$$

The characteristic polynomial, $P(\lambda)$, for the undriven dyad is:

$$P(\lambda) = |\lambda \mathbf{I} - \mathbf{A}|$$

$$= \begin{vmatrix} \lambda & -1 \\ -1 & \lambda \end{vmatrix} \quad \text{.....(3.5)}$$

$$\text{thus, } P(\lambda) = \lambda^2 - 1 \quad \text{.....(3.6)}$$

Number of fundamental cycles = 1 - 2 + 1 = 0

Number of fundamental cutsets = 2 – 1 = 1

3.4.2 Graph Theoretical Description of a Three Bar Open Chain

As a further example of a slightly more complex system, the process is repeated using a three bar open chain with two revolute joints as an example:

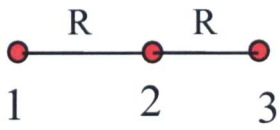


Figure 3.6: Interchange Graph of Three Bar Open Chain

The degree sequence for the interchange graph of a three bar open chain is simply (1,1, 2).

Applying the standard mobility formulae (Equations 3.1 and 3.2) to the three bar open chain, it can be seen that the 2D mobility is $3(3-1) - 2*2 = 2$.

The adjacency matrix, **A**, for the three bar open chain is as follows:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \text{.....(3.7)}$$

The characteristic polynomial, $P(\lambda)$, for the three bar open chain is:

$$P(\lambda) = |\lambda\mathbf{I} - \mathbf{A}|$$

$$= \begin{vmatrix} \lambda & -1 & 0 \\ -1 & \lambda & -1 \\ 0 & -1 & \lambda \end{vmatrix} \quad \text{.....(3.8)}$$

$$\text{Thus, } P(\lambda) = \lambda^3 - 2\lambda \quad \text{.....(3.9)}$$

Number of fundamental cycles = $2 - 3 + 1 = 0$

Number of fundamental cutsets = $3 - 1 = 2$

A more complete presentation of graph theoretical descriptors for basic 2-, 3- and 4-bar systems is contained in Appendix C, Appendix D, and Appendix E.

3.4.3 The Basic Graph Theoretical System Description Developed

It can be seen that graph theoretical methods offer an interesting insight into kinematic system description, but in order to be able to describe more fully the systems under consideration, novel applications of standard graph theory are required. Locomotion subsystem representation needs to be based on a spread of factors in order to achieve the necessary level of faithful and robust representation, and in order to avoid issues of isomorphism. A representation is proposed based on three different aspects of locomotion subsystem design, identified as follows:

- **kinematic topology**

application of 'standard' graph theoretical concepts to kinematic system representation, as discussed in the foregoing sections.

- **joint faults characterisation**

  modification of graph theoretical representations to allow discrimination between different joint types, and their potential fault modes

- **motor / drives characterisation**

  further modification of graph theoretical representations to allow identification of driven and undriven joints, and their power on/off status

A fourth aspect may also be defined, which is **'design principles characterisation'**. This refers to the introduction of methods to allow discrimination based on design features not amenable to analysis by graph theoretical means. An example of this might be whether locomotion relies on the use of wheels or legs/arms, since these two cases are indistinguishable using graph theory. However, this aspect is not pursued in this thesis, in order that the potential of the graph theory approach can be explored more fully.

The issues arising from joint faults characterisation, and motor / drives characterisation are now defined and examined in turn, and the potential of these additional descriptions for enriching system characterisation is evaluated.

3.4.3.1 Joint Faults Representation – the Constraints Matrix

The fault and failure characteristics of a mechanism are strongly influenced by the types of joint involved. These govern the numbers of degrees of freedom (dof) that are constrained or available, and they strongly influence the nature of any faults that occur. In particular, permutations of secondary faults (ie those occurring subsequent to the initial fault) are dictated by the degrees of freedom remaining in the faulty system.

A graph theoretical method is now derived for representing the differences between joint types, and in particular, the differing numbers of degrees of freedom that exist in the nominal, fault free, condition of the various joints.

Consider the simple four-bar linkage shown in Table 3.2, with each edge representing any of the six Reuleaux lower kinematic pairs, j_x, with joint degrees of freedom as shown.

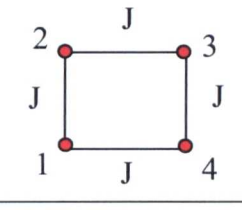
Reuleaux Joint Type (J)	Degrees of Freedom	Linkage Interchange Graph
Revolute	1	
Prismatic	1	
Screw	1	
Cylindric	2	
Planar	3	
Spherical	3	

Table 3.2: Degrees of Freedom of the Six Reuleaux Joints

It was described earlier how this linkage can be represented by an adjacency matrix, **A**, of the following general form, a '1' indicating that a joint exists between the vertices in question.

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \quad \text{..... (3.10)}$$

In order to provide a more detailed method of representing systems, a modification of this adjacency matrix, **A**, to form a 'Constraints Matrix', **CM**, is now proposed, as follows:

$$\mathbf{CM} = \begin{bmatrix} 0 & d_{u1} & 0 & d_{u2} \\ d_{u3} & 0 & d_{u4} & 0 \\ 0 & d_{u5} & 0 & d_{u6} \\ d_{u7} & 0 & d_{u8} & 0 \end{bmatrix} \quad \text{.....(3.11)}$$

where the terms d_{u1} to d_{u8} of the matrix, **CM**, represent the number of **u**nconstrained degrees of freedom within any one of the joints. This will not be the same in all cases, but will be dictated by the joint type at the particular location.

However, this notation might imply that there is no limit on d_u, whereas, for a three-dimensional system, the limit would be six - the number of degrees of freedom that must be specified to fix a rigid body in space. A 'constraints' method is therefore adopted, using the number of constraints imposed on the system elements, such that:

$$d_u + d_c = 6 \quad \text{..... (3.12)}$$

where d_c represents the number of degrees of freedom that are constrained by the joint ie the number of joint constraints.

Thus the constraints matrix, (3.11), becomes:

$$\mathbf{C} = \begin{bmatrix} 6 & (6-d_{u1}) & 0 & (6-d_{u2}) \\ (6-d_{u3}) & 6 & (6-d_{u4}) & 0 \\ 0 & (6-d_{u5}) & 6 & (6-d_{u6}) \\ (6-d_{u7}) & 0 & (6-d_{u8}) & 6 \end{bmatrix} \quad \text{.....(3.13)}$$

In this representation, diagonal elements ('self-zeros') become 6, since no adjacency matrix vertex can have any degrees of freedom relative to itself. Furthermore, those matrix elements representing vertices between which no connections exist ('no-link zeros') continue to be represented as zeros.

The behaviour of this representation in a number of sample cases is now examined by evaluating the characteristic polynomial, $P(\lambda)$, the definition of which has been discussed previously, and is:

$$P(\lambda) = |\lambda \mathbf{I} - \mathbf{C}| \quad \text{..... (3.3)}$$

Thus, from equations (3.3) and (3.13) for the four bar linkage shown in Table 3.2, the following is obtained:

$$P(\lambda) = \begin{vmatrix} (\lambda-6) & -(6-d_{u1}) & 0 & -(6-d_{u2}) \\ -(6-d_{u3}) & (\lambda-6) & -(6-d_{u4}) & 0 \\ 0 & -(6-d_{u5}) & (\lambda-6) & -(6-d_{u6}) \\ -(6-d_{u7}) & 0 & -(6-d_{u8}) & (\lambda-6) \end{vmatrix} \quad \text{.....(3.14)}$$

and assuming revolute, prismatic or screw joints (one dof) in all locations, 3.14 becomes:

$$P(\lambda) = \begin{vmatrix} (\lambda-6) & -5 & 0 & -5 \\ -5 & (\lambda-6) & -5 & 0 \\ 0 & -5 & (\lambda-6) & -5 \\ -5 & 0 & -5 & (\lambda-6) \end{vmatrix} \quad \text{.....(3.15)}$$

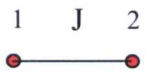
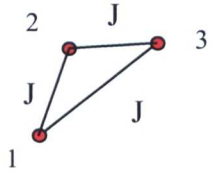
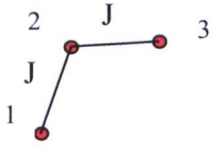
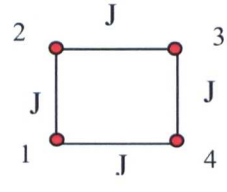
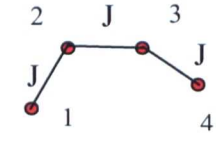
hence, evaluating using Mathcad:

$$P(\lambda) = \lambda^4 - 24\lambda^3 + 116\lambda^2 + 336\lambda - 2304 \quad \text{.....(3.16)}$$

This process is repeated for several typical, simple linkages. For each linkage the characteristic polynomial is derived, and hence the polynomial coefficients, $\{p_0, p_1, p_2 \ldots p_i \ldots p_n\}$ for each of the three groups of joints obtained when ranked according to their degrees of freedom:

- one dof - revolute / prismatic / screw
- two dof – cylindric
- three dof – planar / spherical

Thus the values shown in Table 3.3 are obtained. Note that for the moment, all joints in any one linkage are assumed to have the same number of degrees of freedom.

Linkage Type & Interchange Graph	Type of Representation	Characteristic Polynomial Coefficients					Eigenvalues
		p_0	p_1	p_2	p_3	p_4	
2 Bar 'open' 	'Standard'	1	0	-1			[-1, 1]′
	'Constraint'						
	➢ Revolute / Prismatic / Screw	1	-12	+11			[1, 11]′
	➢ Cylindric	1	-12	+20			[2, 10]′
	➢ Planar / Spherical	1	-12	+27			[3, 9]′
3 Bar 'closed' 	'Standard'	1	0	-3	-2		[1.414, 0, -1.414]′
	'Constraint'						
	➢ Revolute / Prismatic / Screw	1	-18	+33	-16		[13.071, 6, -1.071]′
	➢ Cylindric	1	-18	+60	-56		[11.657, 6, 0.343]′
	➢ Planar / Spherical	1	-18	+81	-108		[10.243, 6, 1.757]′
3 Bar 'open' 	'Standard'	1	0	-2	0		[-1, -1, 2]′
	'Constraint'						
	➢ Revolute / Prismatic / Screw	1	-18	+58	+84		[1, 1, 16]′
	➢ Cylindric	1	-18	+76	-24		[2, 2, 14]′
	➢ Planar / Spherical	1	-18	+90	-108		[3, 3, 12]′
4 Bar 'closed' 	'Standard'	1	0	-4	0	0	[0, -2, 0, 2]′
	'Constraint'						
	➢ Revolute / Prismatic / Screw	1	-24	+116	+336	-2304	[6, 16, 6, -4]′
	➢ Cylindric	1	-24	+152	-96	-1008	[6, 14, 6, -2]′
	➢ Planar / Spherical	1	-24	+180	-432	0	[6, 6, 12, 0]′
4 Bar 'open' 	'Standard'	1	0	-3	0	+1	[0.618, 1.618, -0.618, -1.618]′
	'Constraint'						
	➢ Revolute / Prismatic / Screw	1	-24	+141	+36	-779	[9.09, 14.09, 2.91, -2.09]′
	➢ Cylindric	1	-24	+168	-288	-176	[8.472, 12.472, 3.528, -0.472]′
	➢ Planar / Spherical	1	-24	+189	-540	+405	[7.854, 10.854, 4.146, 1.146]′

Note 1: J refers to a joint that may be any one of the Reuleaux joints specified in Table 3.2, and column 2, above.

Table 3.3: Characteristic Polynomial Coefficients for a Variety of Simple Linkages

As already stated, Yan and Hall [90] observe that the coefficients of the characteristic polynomial for what has earlier been defined as a 'standard' adjacency matrix, **A**, have a specific physical meaning, and can be evaluated by a "Coefficients by Inspection" approach, with the meanings

listed earlier. It is now possible to investigate whether the values derived for the proposed 'constraints' method can be similarly interpreted. It is noted from [Stephenson, 19nn [67]] that the coefficients of a polynomial are related to its eigenvalues by the following relationships:

$$p_0 = 1 \quad \text{..... (3.17a)}$$

$$p_1 = -\Sigma \lambda_i \quad \text{..... (3.17b)}$$

$$p_2 = +\sum_{i \neq j} \lambda_i \lambda_j \quad \text{..... (3.17c)}$$

$$p_3 = -\sum_{i \neq j \neq k} \lambda_i \lambda_j \lambda_k \quad \text{..... (3.17d)}$$

$$p_n = \lambda_1 \lambda_2 \lambda_3 \ldots \lambda_n \ldots\ldots \{\equiv |\mathbf{C}|\} \quad \text{..... (3.17e)}$$

Considering the 'residual constraints' values of the characteristic polynomial coefficients in Table 3.3, it is noted that:

- Leading coefficient, p_o , is always unity
- Second coefficient, p_1, is the negative of the total number of degrees of freedom available in the disconnected system – the negative of the trace of the constraints matrix = -6n (where n is the number of vertices in the interchange graph). By inspection in Table 3.3, it can be seen that this also complies with (3.17b).
- Third coefficient: The derivation of p_2 is not immediately apparent. Evaluate the same set of examples as in Table 3.3, but with zeros on the leading diagonal. This will set the value of coefficient p_1 to zero, ie eliminating it, The results obtained are presented in Table 3.4.

  Let **B** represent this modified constraints matrix, so that:

  $$\mathbf{B} = \mathbf{C} - 6\mathbf{I} \quad \text{..... (3.18)}$$

  If $Q(\lambda)$ is the characteristic polynomial of the modified constraints matrix, then:

  $$Q(\lambda) = |\mu\mathbf{I} - \mathbf{B}| \quad \text{..... (3.19)}$$

  Thus, substituting (3.18) into (3.19):

  $$Q(\lambda) = |\mu\mathbf{I} - \mathbf{C} + 6\mathbf{I}| = |(\mu+6)\mathbf{I} - \mathbf{C}| \quad \text{..... (3.20)}$$

  Therefore, comparing with (3.3), $P(\lambda) = |\lambda\mathbf{I} - \mathbf{A}|$, the relationship between the eigenvalues, λ, of the characteristic polynomial of **C**, and the eigenvalues, μ, of the characteristic polynomial of **B** is inferred to be:

  $$\lambda = \mu + 6 \quad \text{..... (3.21)}$$

Referring to the eigenvalues provided in Tables 3.3 and 3.4, it can be seen that this is, indeed, the case, and it is therefore possible to be confident that a fixed relationship exists, and it is noted from Table 3.4 that the relationship governing p_2 (as derived for **B**) is as follows:

$$p_2 = -P.d_c^2 \quad \text{..... (3.22)}$$

where P is the number of connected pairs in the interchange graph.

Using these facts, it is possible to deduce the p_2 relationship for **C** within Table 3.3. Denote the diagonal matrices with 6 in every diagonal position as follows:

$D6_2$ = diag (6,6), with determinant | $D6_2$ |

$D6_3$ = diag (6,6,6), with determinant | $D6_3$ |

$D6_n$ = diag (6,6 . . . . . 6,6), with determinant | $D6_n$ |

Thus, p_2 for a linkage with constraints matrix of order n is:

$$p_2 := \frac{|D6_n|}{(n-1)!} - P.d_c^2 \quad \text{.....(3.23)}$$

Noting that $|D6_n| = 6^n$, then:

$$p_2 := \frac{\left(6^n - P.d_c^2.(n-1)!\right)}{(n-1)!} \quad \text{.....(3.24)}$$

Thus, there is a similarity with the Yan and Hall approach, since a direct relationship exists between p_2 and the number of connected pairs.

- Fourth coefficient: The relationship for p_3 is not apparent. Proceeding as for p_2, and referring to Table 3.4, only the '3 Bar closed' linkage has values for p_3. If the Yan and Hall interpretation reads-across to this revised method, this is to be expected, since only this one example has a three pair loop. Based on this one example, the relationship for p_3 as derived for **B** is:

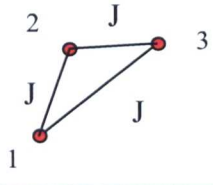
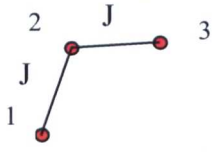
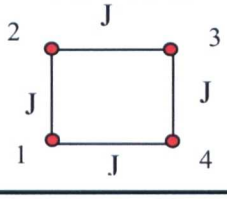
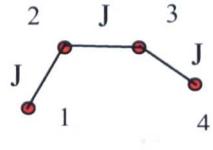
$$p_3 = -2*L_3.d_c^3 \quad \text{..... (3.25)}$$

where L_3 is the number of 'three pair loops' in the interchange graph.

Using this fact, it should now be possible to proceed to derive the relationship for p_3 based on Table 3.3. This is not undertaken at this time, because it is considered more appropriately delayed until later in this thesis when additional information is available.

In summary, therefore, it is considered that the adjacency matrix can be satisfactorily modified to create the **Constraints Matrix**, incorporating representation of joint types, and that this

representation appears at first sight to show the same basic features as discussed by Yan and Hall [90]. However, this result requires further validation once the behaviour of systems under the action of faults is established. This is undertaken in Chapter 5, and the topic of 'coefficients by inspection' revisited in Chapter 8.

Linkage Interchange Graph	Type of Representation	CP Coefficients					Eigenvalues
		p_0	p_1	p_2	p_3	p_4	
2 Bar 'open' 	'Standard'	1	0	-1			[-1, 1]′
	'Constraint'						
	➢ Revolute / Prismatic / Screw	1	0	-25			[-5, 5]′
	➢ Cylindric	1	0	-16			[-4, 4]′
	➢ Planar / Spherical	1	0	-9			[-3, 3]′
3 Bar 'closed' 	'Standard'	1	0	-3	-2		[1.414, -1.414]′
	'Constraint'						
	➢ Revolute / Prismatic / Screw	1	0	-75	-250		[7.071, -7.071]′
	➢ Cylindric	1	0	-48	-128		[5.657, -5.657]′
	➢ Planar / Spherical	1	0	-27	-54		[4.243, -4.243]′
3 Bar 'open' 	'Standard'	1	0	-2	0		[-1, -1, 2]′
	'Constraint'						
	➢ Revolute / Prismatic / Screw	1	0	-50	0		[-5, -5, 10]′
	➢ Cylindric	1	0	-32	0		[-4, -4, 8]′
	➢ Planar / Spherical	1	0	-18	0		[-3, -3, 6]′
4 Bar 'closed' 	'Standard'	1	0	-4	0	0	[0, -2, 0, -2]′
	'Constraint'						
	➢ Revolute / Prismatic / Screw	1	0	-100	0	0	[0, -10, 0,10]′
	➢ Cylindric	1	0	-64	0	0	[0, -8, 0, 8]′
	➢ Planar / Spherical	1	0	-36	0	0	[0, 0, 6, -6]′
4 Bar 'open' 	'Standard'	1	0	-3	0	+1	[0.618, 1.618, -0.618, -1.618]′
	'Constraint'						
	➢ Revolute / Prismatic / Screw	1	0	-75	0	+625	[3.09, 8.09, -3.09, -8.09]′
	➢ Cylindric	1	0	-48	0	+256	[2.472, 6.472, -2.472, -6.472]′
	➢ Planar / Spherical	1	0	-27	0	+81	[1.854, 4.854, -1.854, -4.854]′

Note 1: J refers to a joint that may be any one of the Reuleaux joints specified in Table 3.2, and column 2, above.

Table 3.4: Characteristic Polynomial Coefficients for Constraints Matrices with-Leading Diagonal set to Zero

3.4.3.2 Motor / Drives Characterisation

A method is sought for representing driven joints based on the fact that the degrees of freedom of a driven joint are only present when the drive motor is active. When the motor is inactive, the mechanism acts as if the links articulated by the drive are frozen together. A case also exists where a driven joint may fail 'freewheel' or be allowed to move unconstrained when switched off. These situations are conveniently dealt with using methods applicable to undriven joints in normal operation.

Four alternative graph theoretical methods are considered:

- 'Contracted form when inactive'
- 'Modified Joint Matrix when inactive'
- 'Triangulated form when inactive'
- 'Modified Constraints Matrix when inactive'

In brief, these methods can be summarised as follows:

'Contracted form when inactive' - The interchange graph vertices joined by the driven joint are identified as a single vertex when the drive is inactive.

'Modified Joint Matrix when inactive' - The entries in the Adjacency matrix are modified to 'freeze' joints.

'Triangulated form when inactive'- Introduction of a 'switch link' to triangulate the driven joint when inactive.

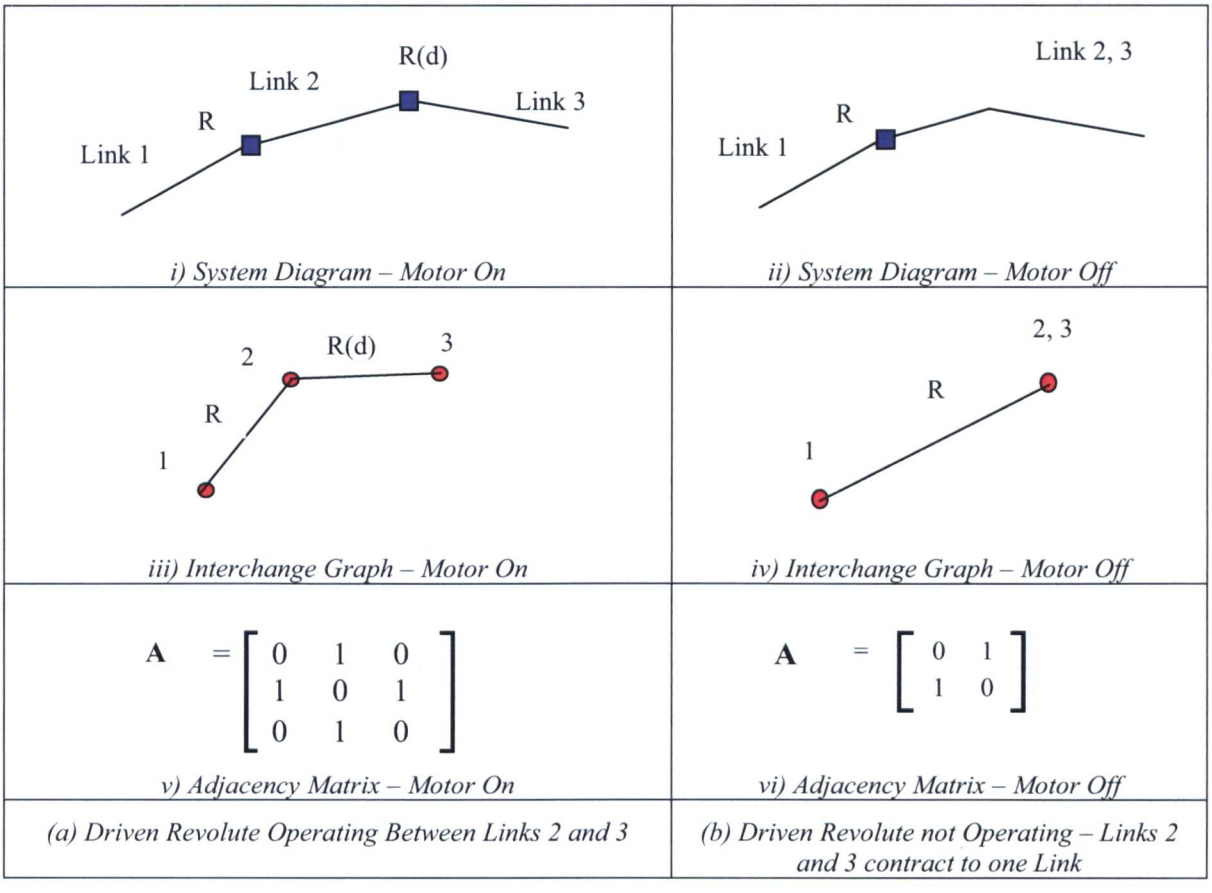
'Modified Constraints Matrix when inactive' - The entries in the constraints matrix are modified to reflect the number of degrees of freedom remaining after drive switch-off.

The following sub-sections now go into greater detail in order to clarify each of these approaches in turn.

3.4.3.2.1 Contracted Form when Inactive

Figure 3.7 shows the system diagram, interchange graph and adjacency matrix of a simple linkage comprising three links, one driven revolute joint, and one undriven revolute joint, in its motor on and motor off states. This is represented using the 'contracted form when inactive' method. When the motor is not operating, associated elements of the mechanism act as if links articulated by the

drive are rigidly joined, if an assumption is made that the motor is non-backdrivable, and free of backlash. In this situation, the relevant part of the Interchange Graph, and the associated section of the Adjacency Matrix revert to a 'contracted' form where the two links articulated by the driven joint combine into a single link.



i) System Diagram – Motor On | *ii) System Diagram – Motor Off*

iii) Interchange Graph – Motor On | *iv) Interchange Graph – Motor Off*

v) Adjacency Matrix – Motor On | *vi) Adjacency Matrix – Motor Off*

(a) Driven Revolute Operating Between Links 2 and 3 | *(b) Driven Revolute not Operating – Links 2 and 3 contract to one Link*

Figure 3.7: Interchange Graph Contraction due to Non-Operating Drive

The primary advantage of this method is that it is simple, and is based on standard graph theoretical methods. However, it can be argued that it is unsuitable for purpose for two reasons:

- The resizing of the Interchange Graph as motors are switched on and off would create difficulties when analysing a mechanism of any complexity.
- The latent ability of the mechanism to lock out the driven joint is not represented, which may have implications for later introduction of fault mode representations

3.4.3.2.2 Modified Joint Matrix when Inactive

A second method which could be used for representing the action of driven joints in the power on and power off conditions, and which would still maintain the Adjacency Matrix order unchanged is now considered. This method relies on the insertion of '1s' into the Adjacency Matrix as shown in

Figure 3.8(vi), which would have the effect of 'triangulating' the joint, and so constrain its mobility as required.

However, when '1s' are inserted in the requisite cells of the Adjacency Matrix, that is to say, the cells which represent the vertices on either side of the driven joint, indicated by the grey background cells in Fig. 3.8(vi), this does not produce the desired effect in the interchange graph. The result of this modification is to triangulate the entire linkage. It is true that in a more complex linkage, this 'locking out' effect would be fairly localised, and not affect the entire linkage, but it would, nonetheless, not be an accurate representation of the real life situation. No way of modifying the adjacency matrix is available that will immobilise one joint and not the other.

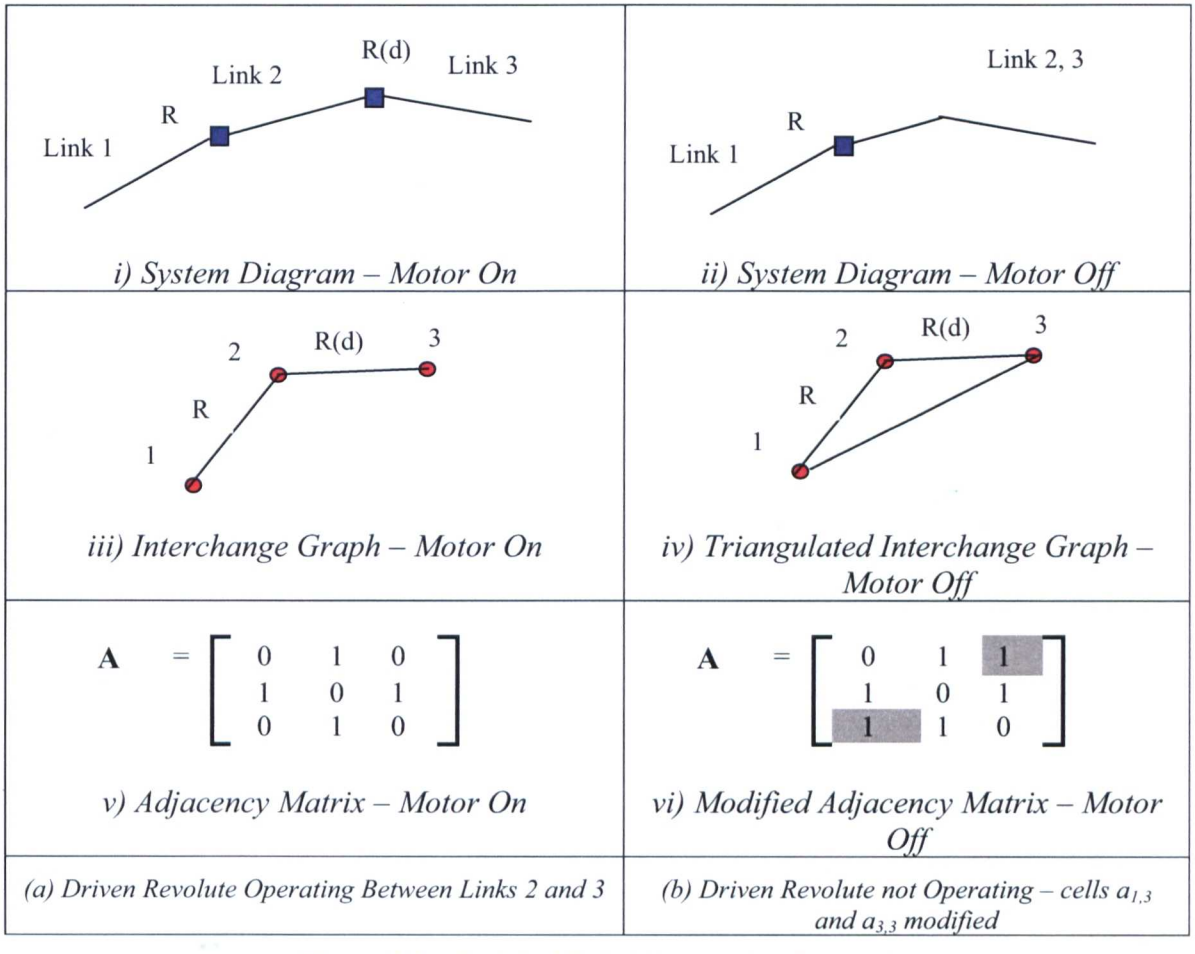


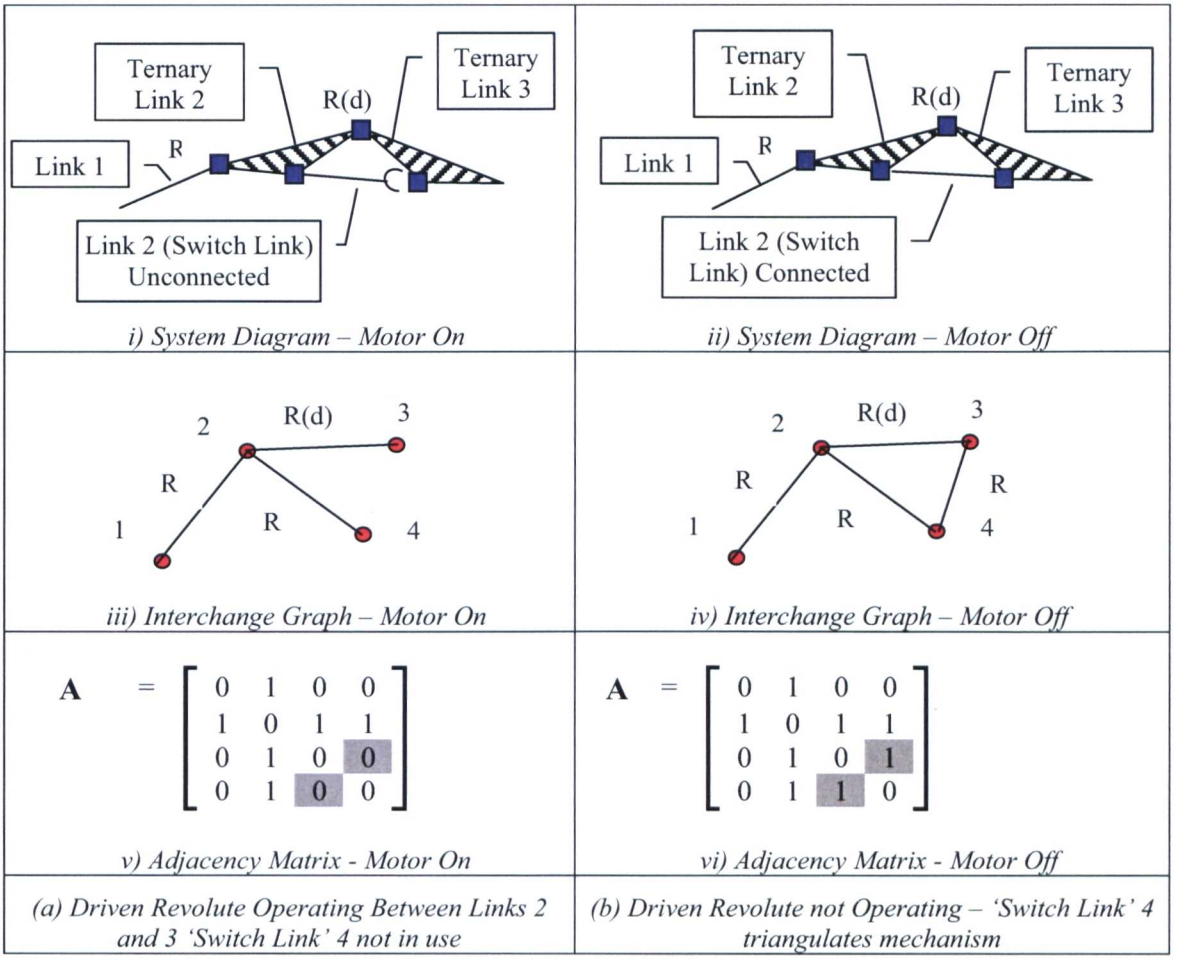
Figure 3.8: 'Modified Joint Matrix when Inactive'

3.4.3.2.3 Triangulated Form when Inactive

A third representation of the motor off condition is considered, based on triangulating the affected part of the linkage by introducing an edge (link) into the Direct Graph (Figure 3.9(i)). Inspection of the direct graphs in the power on and power off conditions shows that, in order to make provision

for this 'switch link', it is necessary to convert links 2 and 3 from binary links to ternary links (as shown by the 'hatched' areas in the two diagrams.

When the motor is on, the 'switch link' has no effect. However, the latent ability to activate or deactivate the drive (by whatever means) is represented by the presence of this unconnected link whilst the motor is running. When the motor is switched off, the 'switch link' is used to triangulate the relevant part of the linkage (Figure 3.9(ii)).



i) System Diagram – Motor On

ii) System Diagram – Motor Off

iii) Interchange Graph – Motor On

iv) Interchange Graph – Motor Off

v) Adjacency Matrix - Motor On

vi) Adjacency Matrix - Motor Off

(a) Driven Revolute Operating Between Links 2 and 3 'Switch Link' 4 not in use

(b) Driven Revolute not Operating – 'Switch Link' 4 triangulates mechanism

Figure 3.9: System Diagram and Interchange Graphs for' Triangulated Form when Inactive', and associated Adjacency Matrices

Referring to Figure 3.9(v) and 3.9(vi), the shaded cells of the Adjacency Matrix change from 0 to 1 according to the status of the drive. Thus, there is the ability to use these elements as 'switchable' elements – a simple matrix manipulation represents the activation or deactivation of drives.

A major advantage of this approach, and the difference between this method and the 'Contracted Form when Inactive' option is that the latter option changes the number of joints within the linkage and does not maintain the order of the Adjacency Matrix constant. This option changes the number of link connections, and so does not suffer from the same drawback. It is considered that this could allow more straightforward representation of systems with multiple driven joints.

Two disadvantages exist, however:

- the use of 'switch links' would increase matrix size and complexity
- it is necessary to justify the introduction of the 'switch link', and to derive a logic supporting the introduction of an artificial joint in the motor off condition.

With regard to the second of these points, it is possible to introduce greater detail into the direct graph to represent better the presence of a motor (Figure 3.10). In this representation, it is possible to argue that the two links represent the rotor and stator of the drive motor, but this is not a very satisfactory explanation, since the rotor and stator of a motor form a cylindric joint, not two unconnected links as would be required in order to make this 'explanation' work. Furthermore, when this revised approach is analysed in the motor on and motor off conditions, it suffers from exactly the same problem as that described in the 'Contracted Form when Inactive' section earlier. This is because when the motor is switched off, and the joint articulated by the motor is triangulated, the two 'switch links' (marked 'motor link #1' and 'motor link #2' in Figure 3.10) become one, therefore the Adjacency matrix has to change order, and nothing has been gained.

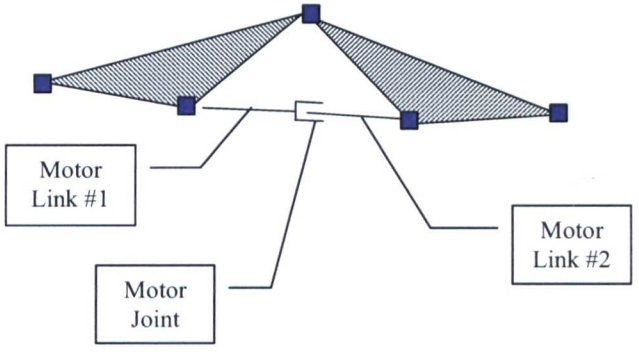


Figure 3.10: System Diagram of a Motor represented using a 'Switch Link'

Therefore, it is considered that this type of representation is, to a large extent, an artificial construct, and it is necessary to look for an alternative solution.

3.4.3.2.4 Modified Constraints Matrix

Referring to work described earlier defining the 'Constraints Matrix', **C**. It is noted that this matrix can also be used to represent motor / drive characteristics in a way compatible with its use for joint representation, and overcoming the drawbacks of the previous three methods suggested for describing motor / drive behaviour.

A 'Motor / Drives (Constraints) Matrix', $\mathbf{C'}$, is specified, such that those cells representing the presence of joints are defined numerically to be the number of constrained degrees of freedom in the joint. Thus, in the **motor on** condition:

$$\mathbf{C'} = \mathbf{C}$$

However, the **motor off** condition is defined as:

$$\mathbf{C'} = \mathbf{C''}$$

$\mathbf{C''}$ contains elements modified to represent the number of degrees of freedom available in the joint after the motor is switched off. In the case of revolute, prismatic, and screw joints, these elements will be 6. For joint types with more degrees of freedom, elements can be 4, 5 or 6 depending both on the joint type, and on the number of degrees of freedom frozen when the motor is switched off, since motor switch off may not freeze all degrees of freedom.

Investigating this approach for a four bar closed linkage, the results shown in Table 3.5 are obtained. These show that the method achieves most objectives, most importantly, it maintains the order of the Adjacency Matrix. Although not analysed at this stage, inspection of the eigenvalues in Table 3.5 shows that direct relationships exist between the eigenvalues representing the various system states.

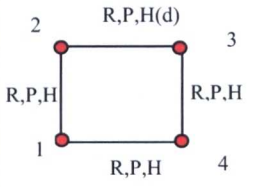
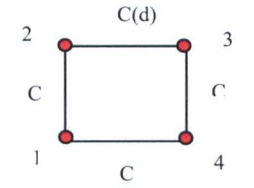
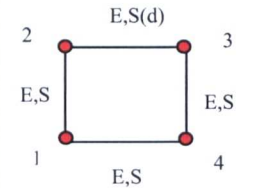
Joint Type and Interchange Graph	Type of Representation	CP Coefficients					Eigenvalues
		p_0	p_1	p_2	p_3	p_4	
	Nominal – Motor On	1	-24	116	336	-2304	[6, 16, 6, -4]′
	Motor Off						
	➢ 1 dof frozen	1	-24	105	468	-2675	[6.475, 5.525, 16.525, -4.525]′
	➢ 2 dof frozen	-	-	-	-	-	-
	➢ 3 dof frozen	-	-	-	-	-	-
	Nominal – Motor On	1	-24	152	-96	-1008	[6, 14, 6, -2]′
	Motor Off						
	➢ 1 dof frozen	1	-24	143	12	-36	[6.877, 5.123, 15.123, -3.123]′
	➢ 2 dof frozen	1	-24	132	144	-128	[5.531, 6.469, -2.531, 14.531]′
	➢ 3 dof frozen	-	-	-	-	-	-
	Nominal – Motor On	1	-24	180	-432	0	[6, 6, 12, 0]′
	Motor Off						
	➢ 1 dof frozen	1	-24	173	-348	-243	[6.459, 5.541, 12.541, -0.541]′
	➢ 2 dof frozen	1	-24	164	-240	-540	[6.838, 5.162, 13.162, -1.162]′
	➢ 3 dof frozen	1	-24	225	-972	1377	[7.146, 4.854, 13.854, -1.854]′

Table 3.5:Characteristics of Motor / Drive Matrices

The main disadvantage of this representation is considered to be that, as with two of the earlier methods, the latent ability of the mechanism to lock out the driven joint is not represented. Nonetheless, this method is proposed for further development.

Now that some basic decisions have been made regarding the representation of joint failures and driven joints, it is possible to proceed to develop system descriptions in greater depth. At this stage, the kinematic modelling methods outlined so far seem likely to provide a suitable basis for definition of the types of system with which this thesis is concerned. A more in-depth look at whether this is so is presented in Chapter 5. However, before this is done, it is necessary to develop a clearer understanding of the scenario within which such representations are intended to be used. For this reason, the next chapter takes a closer look at the nature of the space or spaces within which the system representations are considered to exist, and discusses options which are available for dealing with them.

Chapter 4
The Design Space and Goal Points

Chapter 2 introduced the general space science context within which kinematic systems for planetary exploration vehicles, and in particular, the locomotion sub-systems for these, are required to operate. The chapter included a brief overview of the requirements definition issues which can arise for such systems, and discussed briefly how these requirements - for example, the ability to climb out of trenches and negotiate slopes - can be satisfied by a range of kinematic systems exhibiting differing morphologies, such as legs or variable geometry chassis. Following on from this, Chapter 3 expanded further upon the concepts introduced by Chapter 2, by discussing some of the terrainability objectives, typified by obstacle negotiation under fault conditions, that are key drivers of the mechanical system requirements. It was considered how these can be met by different system morphologies, both in the nominal, fault-free condition, and after the onset of faults. Chapter 3 continued by introducing some fundamental graph theory and linear algebra concepts which could be used to represent some aspects of the behaviour of kinematic systems.

For any design method to be of practical use, it is necessary for it to embody some means whereby potential design solutions can be compared with a set of target requirements, such as those discussed in Chapter 2. The comparison of a number of kinematic systems with such a set of requirements, implies that any potential design solutions and also any target design requirements (both of which may be based on some of the techniques identified in Chapter 3), should ideally both be represented within one common n-dimensional space. Alternatively they may be represented separately in two distinct n-dimensional spaces (one space for the potential designs and another space for the target design) if an appropriate formal (mapping) transformation between the two spaces can be determined.

In this thesis the former approach, involving a single space, is adopted as the more natural representation. This chapter therefore now proceeds to develop the theme of a single design space in greater detail by establishing some of the basic concepts for design points and goal points, and for the space within which they are considered to reside.

4.1 Design Points and Goal Point

It will be shown in later chapters that it is possible to construct a design space comprising design points that represent existing and novel or potential kinematic systems, and that each of these points can be specified by means of a selection of parameters representing some of the kinematic characteristics of the kinematic system it represents.

Additionally, one or more goal points representing desirable solutions for any one given mission may be defined within the same design space. The way in which such goal points may be derived may be through variation of an existing design or designs, or by synthesis based on the required parameters of the system as understood through experience. This theme is developed in greater detail in Section 4.2.

An abstract n-dimensional space is defined by a set of n parameters, and in order for the design points and goal points(s) to co-exist within the same abstract n-dimensional space, it is necessary that both are represented using sets of values corresponding to a common list of (in this case) kinematic parameters.

The choice of parameters used to define the design space is discussed in detail in Section 10.4.8. However, in order to understand the purpose of forthcoming chapters and the way in which these contribute to the development of the design space concept, it is necessary at this stage to develop a preliminary understanding of the representations of kinematic systems that inhabit this space.

The representation of kinematic systems by points within a design space is based on the notion that each 'point' within such a design space is actually a grouping of systems which have a specific relationship to one another. Here, any one 'design point' is defined to consist of a combination of a system's nominal (fault free) condition, together with all of that system's degenerate 'fault states'. This topic is explored in much greater detail in Chapter 8 , where it will be shown how this combination of nominal and 'fault states' (the 'design point') can be viewed and treated as a 'fault

graph' in which the individual graphs for the subject kinematic system's nominal and fault states are embedded, and which represents the available transitions between them. The parameters of a generic fault graph can be used in the construction of both design points and goal points, and these must be consistent with those selected to define the design space.

Thus, the design points and goal point(s) are considered to inhabit the same abstract space, and a visualisation of this is presented as Figure 4.1:.

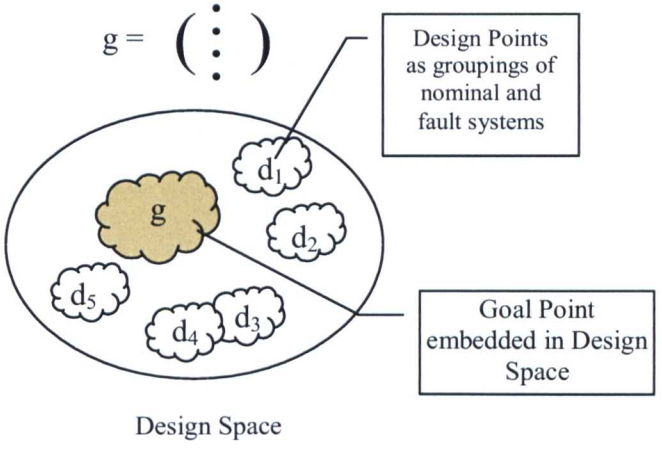


Figure 4.1: Goal Point embedded within a Design Space comprising a Number of Design Points

In general the design space is discrete, consisting of discrete and continuous sub-spaces at various increments along discrete or continuous axes. The space can be considered to be discrete because (as will be seen later in Chapter 10) the parameter sets used for representing the kinematic systems under consideration contain both discrete and continuous parameters. Take, for example, two candidate parameters for selection as part of the design space parameter set. Mobility, M (defined in Chapter 3), is a discrete parameter – it cannot have a non-integer value, whereas entropy, H (discussed in Chapter 7), is a continuous parameter because it can take a range of continuous real values.

Therefore, it can be stated that, in a direction parallel to the axis of any discrete parameter, the space is discrete. In general, the overall design space consists of a countably infinite set of discrete slices (sub-spaces), separated by intervals which are determined by discrete parameter increments (for example, 1, in the case of mobility). Within each discrete slice (sub-space), the space may be continuous, or may have further discrete parameters. In a direction parallel to the axis of any continuous parameter, the space is continuous, and uncountably infinite.

(Note: for the purposes of this thesis, any spaces referred to are specified by mechanical system characteristics only). The notion of sub-spaces (or projections) will be returned to in Section 10.4.8.

4.1.1 Definitions:

Based on the foregoing discussion, the following definitions are established:

Goal Point – A representation in parametric form of the particular attribute values required of a kinematic system fulfilling the identified requirements of a specific mission.

(Mechanical) Design Space – a space defined by parametric representations of kinematic systems appropriate to the planetary exploration task, where "appropriate" is taken to imply kinematic systems created specifically for use within planetary locomotion subsystem development or application programmes.

Design Point – A representation, existing within the Design Space, and expressed in parametric form, of the mechanical attributes of an existing or novel / potential kinematic system.

4.2 Comparing Design Points with Goal Points

Because the goal point(s) and design points exist within the same design space, and are represented in the same way, comparison between them can be carried out using methods based on kinematic and other parameters that will be developed in Chapters 5 to 9.

The goal point(s) may be defined in two main ways. Firstly, through variation of an existing design or designs using specific criteria based on experience. In particular, kinematic system performance improvement may be achieved based on previous designs by changing the value of one or other of the parameters in the goal point. Secondly, the goal point(s) may be synthesised by 'ab initio' selection of the required parameters of the system.

Goal point definition through variation is the preferred method, since this is most closely aligned with real-life design methods. Goal point selection by 'ab initio' synthesis is not preferred because it requires a degree of insight into what combinations of kinematic and other parameters constitute a viable system that may be hard to achieve in practice.

Comparison between an individual goal point or points and one or more design points is achieved by spreadsheet evaluation of one of several different metrics chosen to define an n-dimensional

'inter-system distance' (ISD) between pairs of kinematic systems. Some of the available metrics on which ISD may be based will be discussed in Chapter 6, and any of these may be used. The closest match to a goal point may be established by measuring the ISD between each of the design points and that goal point, and selecting the design point(s) having the minimum ISD between themselves and this goal point.

An alternative approach is to expand one or more of the goal point parameters from a single value into a range of values so that goal point 'boundaries' ('extremal bounds') are established which bound all the parameter values of at least one design point. This approach will be developed in greater detail in Chapter 10.

Selection of design points on the basis of minimum ISD from the goal point(s) is the intuitive choice, but it suffers from the disadvantage that whilst some parameter values for the design point may match the goal point values closely and result in a 'best available match' for those parameters, the values of parameters not matching the goal point values may possibly lie remotely outside the range desired.

The approach to selection summarised above is discussed in greater detail in Chapter 10, and the evaluation of example metrics is demonstrated in Chapter 11.

PART II

THE MECHANICAL DESIGN SPACE

Chapter 5
The Mechanical Design Space

It was shown in Chapter 3 how a combination of Graph Theoretical and Linear Algebra methods provides useful tools for the representation of kinematic system topology. This chapter now continues to develop this proposed modelling / representation methodology further, and to demonstrate that it is possible to use this to generate the necessary richer insights into system robustness, creating representations appropriate for use within the 'Mechanical Design Space'. Arguments are presented here that are central to the representation of mechanical faults in kinematic systems, and, in particular, introducing the key concept of the 'fault path'.

The 'fault path' concept is important because it provides the means of representing the degeneration of a system's kinematic capabilities under the action of faults, doing this by depicting as paths the transitions between the different 'fault states' so created for any one kinematic system. The 'nominal state' of any kinematic system – that is, its fault-free condition – may have the potential to degrade along one or more 'fault paths', and this aggregation of 'fault paths' is referred to as a 'fault graph'. Selection of a methodology for representing these concepts is critical, since this allows description of the way in which one system becomes another as it fails, represents the relationships between all the various states, and can potentially be used to identify routes by which it may be possible to regain system function.

Section 5.1 introduces this chapter by describing the pictorial conventions adopted in order to represent kinematic system structure. This is necessary in order to be able to visualise and describe the structure of the individual 'fault states' of any one kinematic system, and also the relationships between the groupings of these 'fault states' that define a system's 'fault paths'

Section 5.2 introduces in detail how 'fault paths' are constructed from sets of 'fault states' related to one another by mechanical faults which are represented as variations in the degrees of freedom available to the system. The section illustrates, using simple examples, how such mechanical faults can be shown to cause either rotation or reflection of the system eigenvectors. This section introduces a further, important point. As kinematic systems degrade, the 'dimension' of the system, that is the number of vertices (ie links) required to represent the system, will periodically reduce as dictated by the remaining degrees of freedom and the corresponding constraints. The method by which these changes in 'dimension' of a kinematic system are represented is also described here.

For the reason that graph numbering conventions are largely arbitrary, it is important to understand the effect that these can have on any representations being used. Hence, **Section 5.3** discusses the influence that the numbering of a kinematic system's interchange graph vertices can have on the behaviour of 'fault state' representations. An analytical method for determining reflection or rotation of eigenvectors is used to explore the available numbering permutations for several examples, and their underlying structure is discussed with particular reference to this eigenvector rotation and reflection. This discussion of eigenvector permutations introduces the concept of using the terms of the symbolically expanded determinant of a kinematic system's interchange graph constraints matrix to create a description of individual 'fault states'.

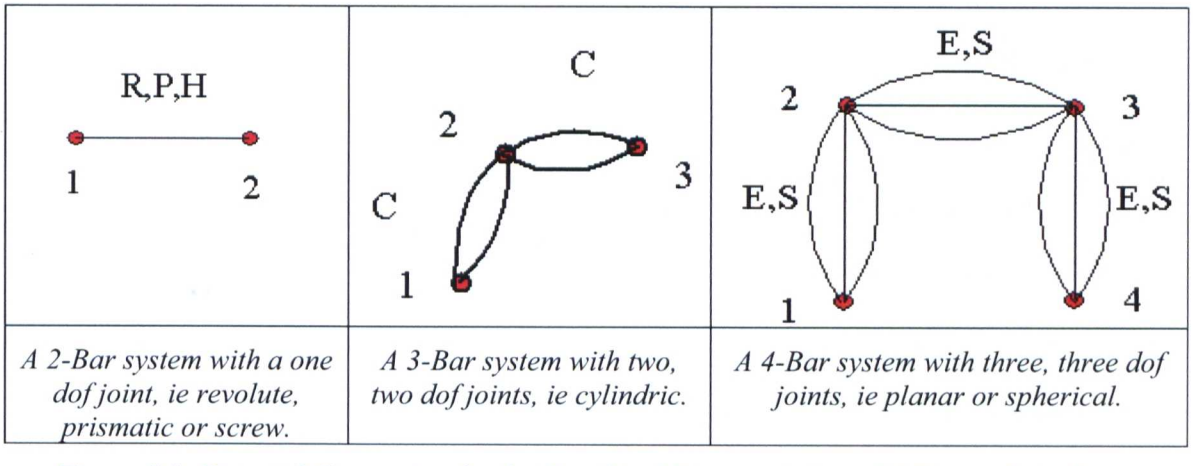
Section 5.4 investigates the potential of using a kinematic system's eigenvalues as a means of representing 'fault paths' and discusses in greater detail the issue introduced in Section 5.1 of changes in system 'dimension' as faults are applied to a kinematic system's nominal state. It is shown that there is potential for visualising these paths in geometric terms, for example lines, planes and hyperplanes.

Determination of any geometrical relationships underlying 'fault path' structure could add significantly to the value of the method proposed, and so **Section 5.5** concludes the chapter with a brief investigation into the geometry of the 'fault paths' used as examples earlier in the chapter. However, a preliminary investigation shows that any geometrical relationship that may exist is not immediately apparent.

5.1 Pictorial Convention for Fault Paths

Using linear algebra, it was shown in Section 3.4.3 how a Characteristic Polynomial may be defined for a graph based on its Adjacency Matrix, **A**, and that such a graph represents aspects of the topology of a kinematic system. It was also established that replacing **A** with a 'Constraints Matrix', **C**, provided an improved representation of the characteristics of the system joints, leading to a revised Characteristic Polynomial, $P(\lambda)$, based on **C**. Once this revised Characteristic Polynomial is constructed, its eigenvalues and eigenvectors can be derived, and these contribute significantly to an improved representation of kinematic system behaviour. In order to achieve the latter, the graph theoretical representations of some of the systems discussed previously are examined in greater detail in the next section. However, since the concept of the Constraints Matrix, **C**, was introduced without developing a pictorial convention for representing the different joint types in the system interchange graphs, it is appropriate to rectify this before continuing further. Section 3.4.3 also introduced basic ideas for the representation of faulty joints, and for driven joints, and the pictorial convention described is also applicable to these.

For the purposes of the discussion here, an interchange graph pictorial convention is adopted which represents each of the joints connecting a pair of graph vertices by a number of edges corresponding to the number of degrees of freedom appropriate to the joint type under consideration. This approach is illustrated in Figure 5.1, for three simple example systems.



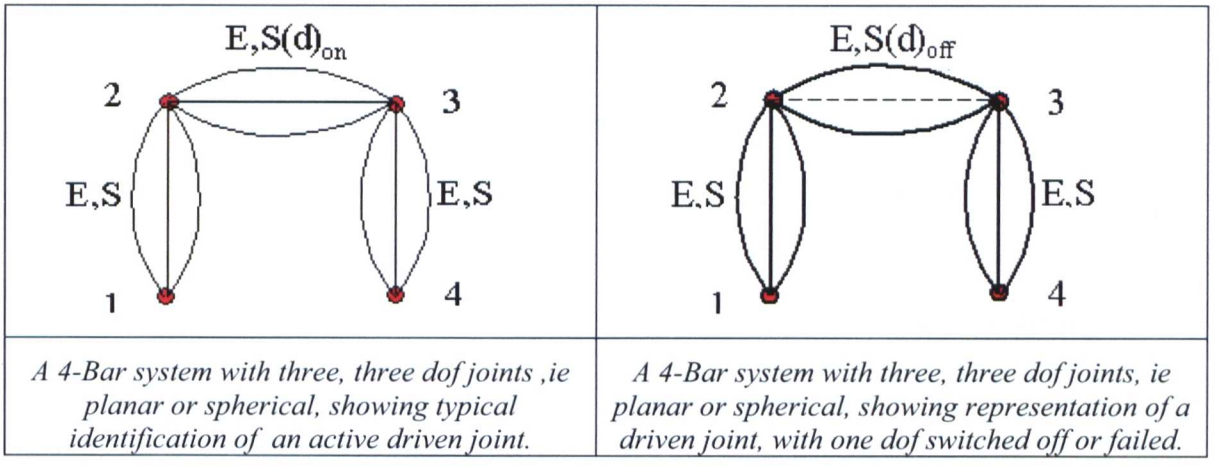
A 2-Bar system with a one dof joint, ie revolute, prismatic or screw.	*A 3-Bar system with two, two dof joints, ie cylindric.*	*A 4-Bar system with three, three dof joints, ie planar or spherical.*

Figure 5.1: Pictorial Convention for the Graphical Representation of Different Joint Types

A more comprehensive dataset of example systems – hereafter 'the dataset' - of 2, 3 and 4-Bar system representations, together with a key showing the system reference number methodology applied throughout this thesis, is provided in Appendix C, Appendix D, and Appendix E.

The pictorial convention may be further extended by using it to indicate which joints are driven, or lose freedoms, or become locked. The effect of having a driven joint actively operating in a system is that the degree of freedom controlled by the drive remains available to the system, and so the graph remains unchanged. When the drive is switched off, the degree of freedom provided by the drive is removed temporarily from the system (but may be reinstated by switching the drive on again). In the event that the drive fails, the degree of freedom provided by the drive is removed permanently from the system.

The pictorial convention described above is extended so that when a degree of freedom of a joint is lost, this is shown by replacing a full edge with a dotted edge in the graph, as shown in Figure 5.2. Additionally, where a joint is being actively driven, this is annotated '$(d)_{on}$', and in circumstances where a driven joint exists, but is inactive (by choice, or through a fault), this is shown as '$(d)_{off}$'. This is also shown in Figure 5.2.



A 4-Bar system with three, three dof joints ,ie planar or spherical, showing typical identification of an active driven joint.	*A 4-Bar system with three, three dof joints, ie planar or spherical, showing representation of a driven joint, with one dof switched off or failed.*

Figure 5.2: Pictorial Convention for the Graphical Representation of Systems with Driven Joints

Even under the normal operating conditions of a nominal - that is to say, unfailed - system incorporating driven joints, there may well be a continual variation from moment to moment of the representative interchange graph, according to the exact nature of the motor's behaviour as it executes its normal mode of operation. This will certainly be the case for robotic systems which will be continually executing movements to alter the pose and posture required to achieve their objectives.

5.2 Fault Path Behaviour of Eigenvectors

5.2.1 Fault States With and Without 'Loss of Edge'

Before developing the concept of the 'Fault Path', additional terminology is introduced to describe the way in which systems undergoing progressive faults are represented.

When considering how to represent edges in fault states, it can be seen that two possible conditions can arise. The first of these is those situations, either where the loss of edge is reversible, such as is the case with the operation of a motor switching on and off, or where the loss of only one of several degrees of freedom between vertices has occurred – that is, where relative motion of the two vertices in question remains possible. This case is referred to as being *'without loss of edge'*. In this case, the graph retains the original number of vertices. The second instance refers to those situations where the change is irreversible and leads to the two vertices in question becoming locked together – in effect becoming the same vertex. This case is referred to as *'node identification with loss of edge'*.

5.2.2 The Fault Path Defined

To examine further how to represent fault configurations, consider the four systems identified in Figure 5.3, where the pictorial convention introduced above has been used. Any given sequence of faults applied to an initial system configuration, will cause it to degrade sequentially into its associated fault states. This is shown in Figure 5.3 for the case where a nominal three-bar system with planar or spherical joints degrades progressively into systems with reduced degrees of freedom (reference numbers are those defined in Appendix D).

The fault path can be considered to comprise a sequence of discrete steps or states, each corresponding to loss of a degree of freedom within one of the joints. Such a fault path can be considered to continue through a number of degenerate states until one is reached which represents a terminating condition. Such a terminating condition would be a state where all system movement has been lost, or where the movement that remains cannot be utilised in any meaningful way.

Even for simple systems, there are significant numbers of fault permutations available. In order to simplify the discussion so that key issues can be identified and discussed, the faults in the example system are considered to be confined to one joint only. Additionally, for ease of visualisation, each fault is considered to arise as a discrete step, although, in practice, this may not be the case, or the steps may follow one another so rapidly that to all intents and purposes they are simultaneous. In Figure 5.3, a 'without loss of edge' treatment is adopted, as defined previously.

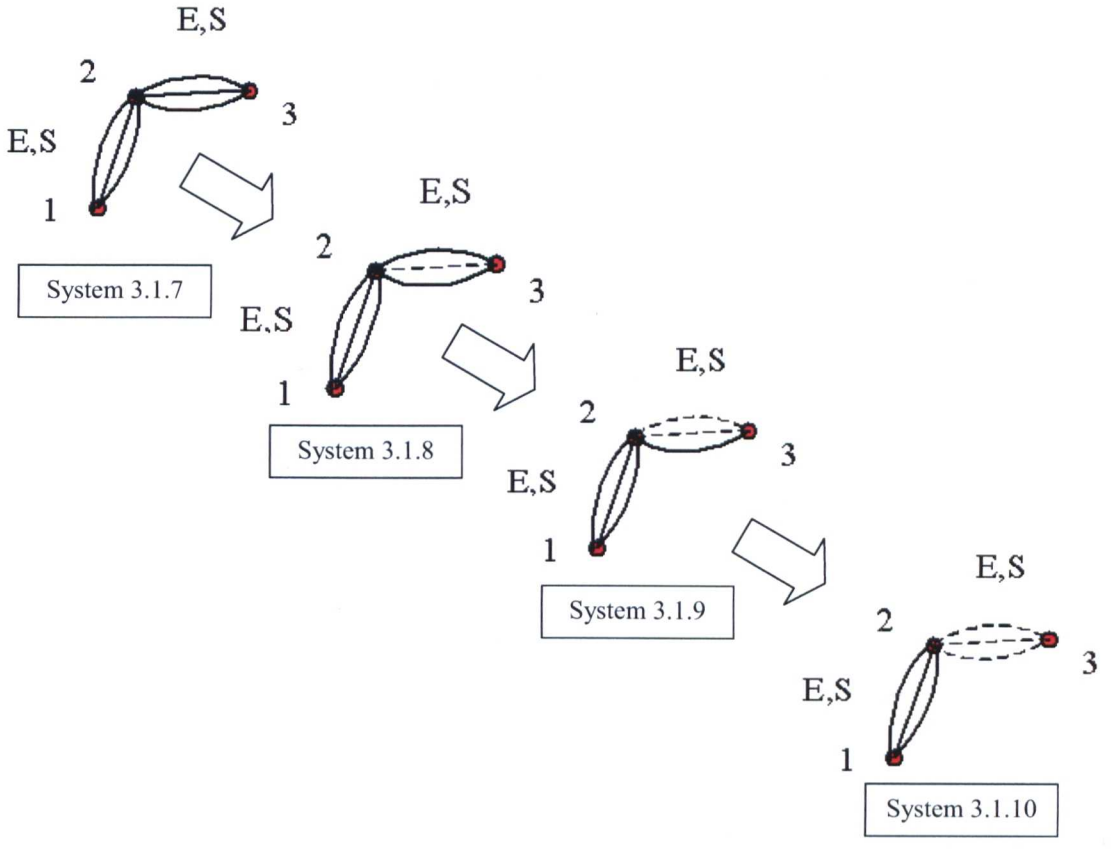


Figure 5.3: Degeneration of a 3 Bar System under the action of Progressive Faults

Additional information regarding the nature of the degeneration illustrated above can be obtained using the eigenvectors of the four systems illustrated in Figure 5.3. This information is presented in Table 5.1, which relates directly to the four systems in Figure 5.3. In the table, the diagonals of the four matrices in the upper row identify the eigenvalues of the four systems, and the eigenvectors form the columns of the lower row of four matrices.

System 3.1.7	System 3.1.8	System 3.1.9	System 3.1.10
Eigenvalues, D	**Eigenvalues, D**	**Eigenvalues, D**	**Eigenvalues, D**
$\begin{bmatrix} 10.243 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 1.757 \end{bmatrix}$	$\begin{bmatrix} 6 & 0 & 0 \\ 0 & 11 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 6 & 0 & 0 \\ 0 & 11.831 & 0 \\ 0 & 0 & 0.169 \end{bmatrix}$	$\begin{bmatrix} 6 & 0 & 0 \\ 0 & 12.708 & 0 \\ 0 & 0 & -0.708 \end{bmatrix}$
Eigenvectors	**Eigenvectors**	**Eigenvectors**	**Eigenvectors**
$\begin{bmatrix} 0.5 & 0.707 & 0.5 \\ 0.707 & 0 & -0.707 \\ 0.5 & -0.707 & 0.5 \end{bmatrix}$	$\begin{bmatrix} -0.8 & 0.424 & 0.424 \\ 0 & 0.707 & -0.707 \\ 0.6 & 0.566 & 0.566 \end{bmatrix}$	$\begin{bmatrix} -0.857 & 0.364 & 0.364 \\ 0 & 0.707 & -0.707 \\ 0.514 & 0.606 & 0.606 \end{bmatrix}$	$\begin{bmatrix} 0.894 & -0.316 & 0.316 \\ 0 & -0.707 & -0.707 \\ -0.447 & -0.632 & 0.632 \end{bmatrix}$

Table 5.1: Eigenvalues and Eigenvectors of Four Typical 3-Bar Systems with Progressive Faults

In Table 5.1, the eigenvectors are normalised, and **D** is a diagonal 3x3 real square matrix. The matrix D is obtained from the constraints matrix, **C**, by using a 'similarity transformation', **D** = **ŨCU**. From linear algebra, it is well known that 'a real square symmetric n x n matrix, **C**, with distinct or repeated eigenvalues, may be diagonalised by an orthogonal transformation **D** = **ŨCU** where **U** is the orthogonal matrix whose columns are formed from a set of **orthonormal eigenvectors** of **C**, and **Ũ** is the transpose of **U**. The diagonal matrix **D** so formed has the eigenvalues of **C** as its elements. An orthogonal matrix is one for which its inverse is equal to its transpose' [54]. In this case, **C** is the Constraints Matrix of the system interchange graph).

As a system progresses through a sequence of fault states, the latter give rise to a sequence of constraint matrices (one for each step of the sequence). Figure 5.4 shows an example of the behaviour of the orthonormal eigenvectors corresponding to these constraints matrices, as applicable to the fault path illustrated in Figure 5.3. It can be seen that the effect of each system fault is to cause either a rotation or a reflection of the eigenvectors, because they continue to remain orthonormal. In other words, the eigenvectors for any one 3D system state should be viewed as **a triad of eigenvectors, the behaviour of which can be considered analogous to that of a rigid body and its mirror image.** (Note also that, in the diagrams, the x, y and z axes are not significant, and only the relative angular relationships between the sets of eigenvectors is meaningful). For

higher order systems, then the triad of eigenvectors will be replaced with a 'rigid' orthonormal system ('n-ad') of four or more eigenvectors.

Further examples of similar fault paths are given in Figures 5.5 and 5.6. Figure 5.5 shows the degradation of a three vertex system containing cylindric joints – that is to say, a system where there are two dof between each of the vertices. Figure 5.6 shows the case of a three vertex system where the joints are either revolute, prismatic or screw – that is to say, systems where there is only one dof between the vertices.

Before presenting the figures referred to above, it should be emphasised that these are instances of the representations to illustrate the fault path concept under consideration, and only represent subsets of the possible eigenvector arrangements. The orientation of the eigenvectors is not only affected by the particular fault under consideration, but also by the original allocation of vertex labels used when deriving the initial constraints matrices. Thus, there are a number of permutations of eigenvector labels that need to be considered in deriving any underlying relationships, and these are discussed further in 5.3.

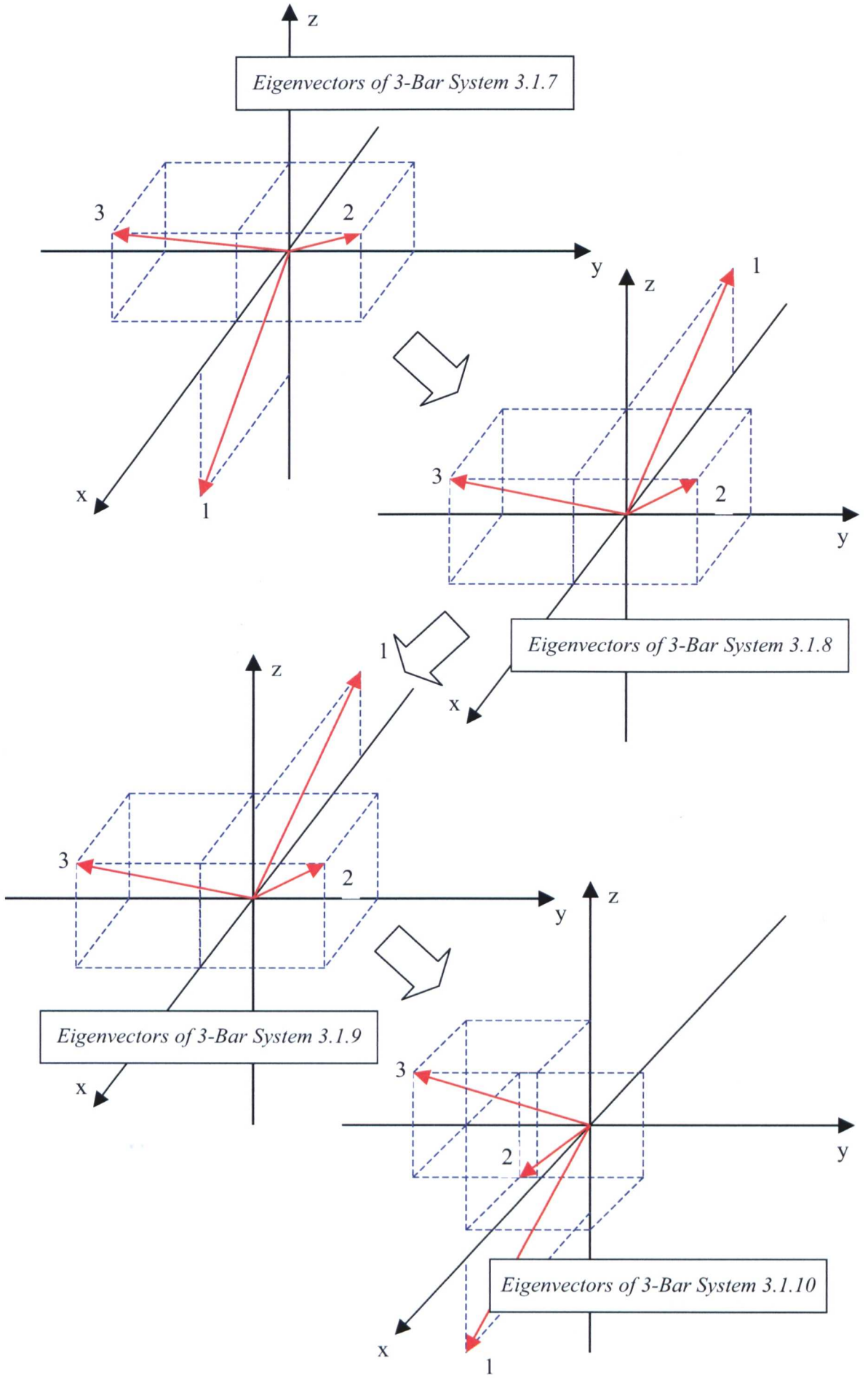


Figure 5.4:Evolution of 3 Bar, 3 dof, System Eigenvectors with Fault Progression

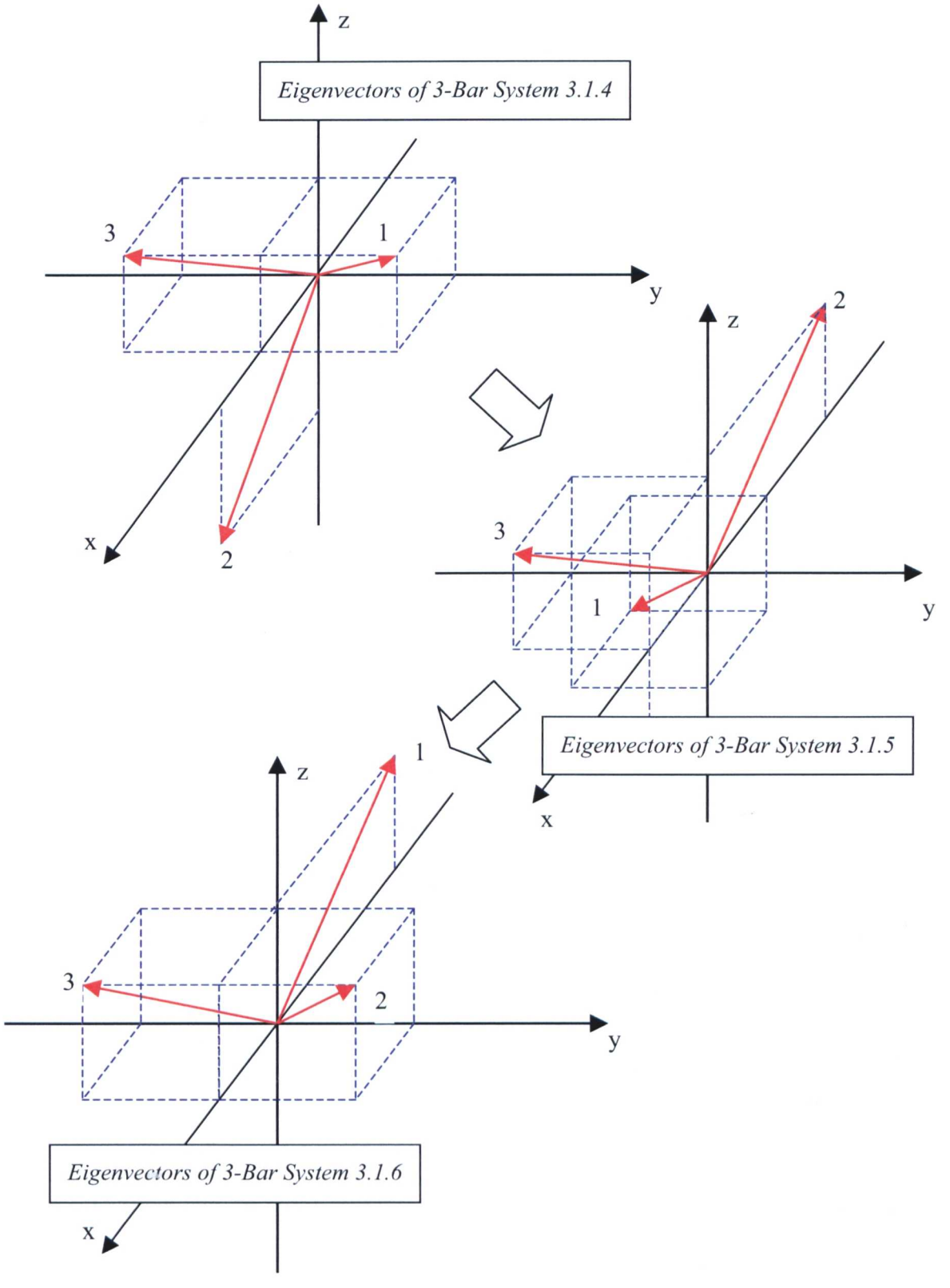


Figure 5.5:Evolution of 3 Bar, 2 dof, System Eigenvectors with Fault Progression

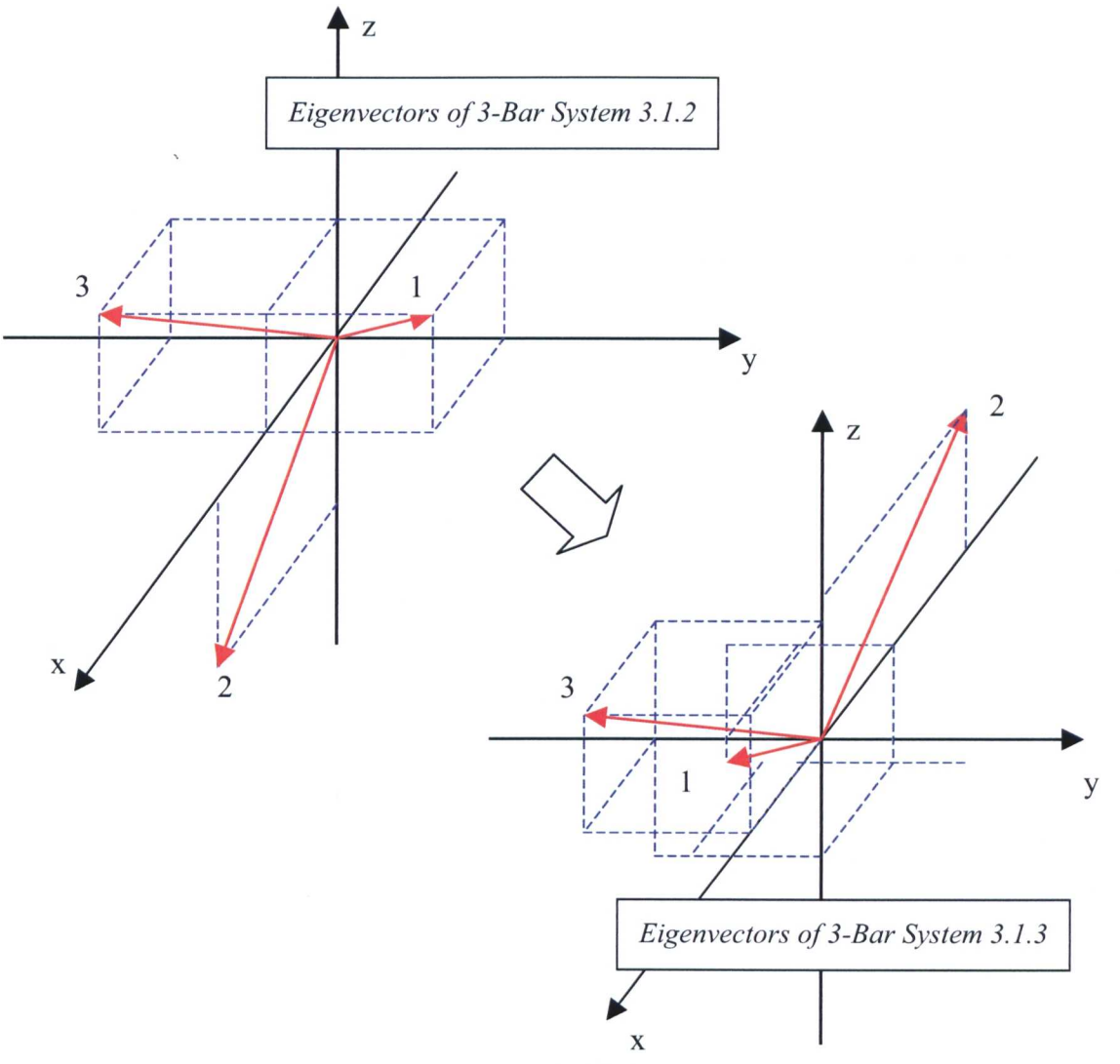


Figure 5.6:Evolution of 3 Bar, 1 dof, System Eigenvectors with Fault Progression

It can be seen that Figures 5.4, 5.5 and 5.6 show fault paths that all end with a system which has no degrees of freedom between the vertices on either side of the failed joint. Referring to the discussion earlier in this chapter on how to treat loss of freedoms between vertices, it will be seen that the system faults are represented as being 'without loss of edge' – that is to say that the lost degrees of freedom are considered to be recoverable – for example, by switching a motor on again. Alternatively, these fault cases could be considered unrecoverable, in which case they could be considered to fall into the 'node identification with loss of edge' category. If this latter approach were taken, this would imply that the system states at the ends of the fault paths shown should be represented as two-bar systems.

Considering the two-bar systems in Appendix C, it can be seen that they have the same eigenvector form of two orthonormal vectors as shown in Figure 5.7.

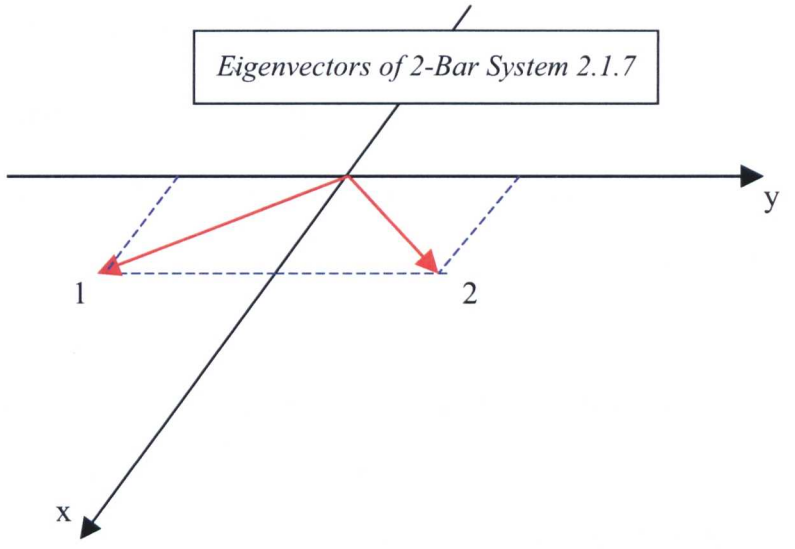


Figure 5.7:The Universal form of 2-Bar Eigenvectors

The matrices corresponding to the systems at the ends of each of the fault paths, have the eigenvectors shown below. The shaded vector lies in the xz plane at y = 0. When the system collapses from a 3-Bar to a 2-Bar system, the y plane disappears, and so the shaded vector will disappear with it, leaving only the x and z values to be considered.

$$\begin{bmatrix} -0.453 & -0.768 & 0.453 \\ -0.707 & 0 & -0.707 \\ -0.543 & 0.64 & 0.543 \end{bmatrix} \quad \ldots\ldots \quad \text{System 3.1.3 (1 dof)}$$

$$\begin{bmatrix} -0.832 & 0.392 & 0.392 \\ 0 & 0.707 & -0.707 \\ 0.555 & 0.588 & 0.588 \end{bmatrix} \quad \ldots\ldots \quad \text{System 3.1.6 (2 dof)}$$

$$\begin{bmatrix} 0.894 & -0.316 & 0.316 \\ 0 & -0.707 & -0.707 \\ -0.447 & -0.632 & 0.632 \end{bmatrix} \quad \ldots\ldots \quad \text{System 3.1.10 (3 dof)}$$

By applying Pythagoras' theorem, it can be shown that the hypotenuse of the right-angled triangles defined by the orthonormal x and z vectors is always 0.707, and hence it can be seen that the orthonormal vectors of the 2D systems are present:

$$\begin{bmatrix} 0.707 & 0.707 \\ -0.707 & 0.707 \end{bmatrix} \quad \ldots\ldots \quad \text{2-Bar Systems}$$

but orientated differently for each system. Thus the two concepts of 'node identification with loss of edge' and 'node identification without loss of edge' are compatible in those situations where all dof between two vertices have been lost. The equivalence of the 3-Bar and 2-Bar representation of

the same condition demonstrates that moving 'across dimensions' as part of a fault path degradation sequence is logically supportable.

5.3 Detailed Discussion of Eigenvector Permutations

5.3.1 Issues Arising from Vertex Numbering Permutations

As stated earlier in this chapter, it is not meaningful to comment in detail on the behaviour of system eigenvectors during progressive fault-induced degradation, without first examining these in relation to the possible permutations of vertex numbering. In many instances, the effects of vertex labelling choice can be ignored, since the techniques discussed are frequently either independent of the vertex labelling chosen, or the labelling can be ignored provided that consistency is maintained. However, in the situation being considered here, this is not the case. Here, the labelling system is significant because it affects the orientation of the eigenvectors, and so different results will be obtained, depending on the labelling adopted. Consequently, there is a danger that the inferences drawn from the eigenvector behaviour may be misleading. It is therefore necessary to investigate in detail the effect that such choices can have.

Consider further the example system fault paths shown in Figures 5.4, 5.5 and 5.6, that is to say, three-bar open systems with their vertices connected with either 3 dof, 2 dof or 1 dof joints. The number of permutations, P_n, is given by the expression $P_n = n!$, where n is the number of items under consideration, and the permutations can be effected either by an even number or an odd number of transpositions (interchange of two) starting from an initial ordering. Even (odd) permutations are related to the initial ordering by an even (odd) number of transpositions [40]. Therefore, for the three-bar systems under consideration here, the permutations of vertex numbering available are as shown in Table 5.2.

Permutation No.	Vertex Order	Even / Odd
1	1-2-3	Even
2	2-1-3	Odd
3	2-3-1	Even
4	3-2-1	Odd
5	3-1-2	Even
6	1-3-2	Odd
1	1-2-3	Even

Table 5.2: Even and Odd Permutations of Vertex Labelling

If the eigenvectors for the systems and fault paths illustrated in Figures 5.4, 5.5 and 5.6 are now re-examined for each of the possible vertex labelling permutations identified above, the results presented in Figures 5.8, 5.9 and 5.10 are obtained.

Figure 5.8 shows each of the fault states in the fault path shown in Figure 5.4 for three-bar systems connected with planar or spherical (three dof) joints. The figure is arranged with the rows representing the fault states for some chosen initial vertex labelling. The same fault path is then repeated in each row down the page for each vertex labelling permutation identified in Table 5.2.

Figures 5.9 and 5.10 use the same layout as for Figure 5.8 to show the eigenvector behaviour of three-bar open systems connected with either cylindric (two dof – Figure 5.5) or revolute, prismatic or screw (one dof – Figure 5.6) joints.

The figures also identify whether the transition to the next fault state across the page is achieved by means of a rotation, or a reflection of the triad of eigenvectors, and this topic is now examined in greater detail.

5.3.2 Eigenvector Rotation and Reflection

As already remarked, the transition from one fault state to another along any fault path involves either a rotation or a reflection of the triad of orthonormal eigenvectors. Any adjacent pair of fault states constitutes a single 'degradation' induced by a change in the number of degrees of freedom in the system. The relationship between the sets of eigenvectors for each of the two fault states can be examined to identify whether the degradation involves a reflection or a rotation.

Rooney [59] outlines a method such that a matrix

$$R = \begin{bmatrix} r11 & r12 & r13 \\ r21 & r22 & r23 \\ r31 & r32 & r33 \end{bmatrix}$$

will exist that transforms the initial three (orthonormal) eigenvectors e1, e2, e3 into the next three (orthonormal) eigenvectors e'_1, e'_2, e'_3, in the degradation sequence along a fault path. Therefore:

$$\mathbf{R}\,\mathbf{e}_1 = \mathbf{e}'_1 \qquad \mathbf{e}_1 = (e_{1x}, e_{1y}, e_{1z})^T \qquad \mathbf{e}'_1 = (e'_{1x}, e'_{1y}, e'_{1z})^T$$

$$\mathbf{R}\,\mathbf{e}_2 = \mathbf{e}'_2 \quad \text{where} \quad \mathbf{e}_2 = (e_{2x}, e_{2y}, e_{2z})^T \quad \text{and} \quad \mathbf{e}'_2 = (e'_{2x}, e'_{2y}, e'_{2z})^T$$

$$\mathbf{R}\,\mathbf{e}_3 = \mathbf{e}'_3 \qquad \mathbf{e}_3 = (e_{3x}, e_{3y}, e_{3z})^T \qquad \mathbf{e}'_3 = (e'_{3x}, e'_{3y}, e'_{3z})^T$$

The matrix **R** may represent a rotation or a reflection. If **R** represents a rotation, then determinant

$$\det(\mathbf{R}) = \begin{vmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{vmatrix} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5.1)$$

will have the value det(**R**) = +1, and if the matrix R represents a reflection, then its determinant will have the value det(**R**) = – 1.

The matrix **E** is defined as the transpose of the matrix of the initial three eigenvectors arranged as the columns, assuming that these are normally stated with x, y and z co-ordinates arranged vertically (which is the convention adopted by Mathcad). Therefore:

$$\mathbf{E} = \begin{bmatrix} e_{1x} & e_{1y} & e_{1z} \\ e_{2x} & e_{2y} & e_{2z} \\ e_{3x} & e_{3y} & e_{3z} \end{bmatrix}$$

It can be shown that:

$$\mathbf{E}\,\mathbf{r}_1 = \mathbf{e}'_x$$

$$\mathbf{E}\,\mathbf{r}_2 = \mathbf{e}'_y$$

$$\mathbf{E}\,\mathbf{r}_3 = \mathbf{e}'_z$$

Where $\mathbf{r}_1$, $\mathbf{r}_2$, $\mathbf{r}_3$ are the first, second and third columns of **R**. So, these are obtained by finding the inverse $\mathbf{E}^{-1}$ of the matrix **E**, and then:

$$\mathbf{r}_1 = \mathbf{E}^{-1}\,\mathbf{e}'_x$$

$$\mathbf{r}_2 = \mathbf{E}^{-1}\,\mathbf{e}'_y$$

$$\mathbf{r}_3 = \mathbf{E}^{-1}\,\mathbf{e}'_z$$

This process gives the nine elements of the rotation / reflection matrix **R** and the value (± 1) of the determinant det(**R**) of **R** then resolves whether **R** is a rotation or reflection, as stated previously.

The following sub-section presents one example of the application of this approach to some sample data:

5.3.2.1 Three Bar Open Systems with Planar or Spherical Joints, Vertices Numbered 1-2-3 - First Degradation

Based on the fault path shown in Figure 5.3, and extracting the relevant data from Appendix D, the method identified above yields the following results for the first degradation of the first permutation:

Initial State Eigenvectors (system 3.1.7): Degraded State Eigenvectors (system 3.1.8):

$$\mathbf{S} = \begin{bmatrix} 0.707 & 0.5 & 0.5 \\ 0 & 0.707 & -0.707 \\ -0.707 & 0.5 & 0.5 \end{bmatrix} \qquad \mathbf{S'} = \begin{bmatrix} -0.8 & 0.424 & 0.424 \\ 0 & 0.707 & -0.707 \\ 0.6 & 0.566 & 0.566 \end{bmatrix}$$

where the matrix **S**, is the matrix of eigenvectors output by Mathcad for the initial system configuration, and **S'** is the matrix of eigenvectors output by Mathcad for the changed (ie degraded) system configuration. Note that, in order for the eigenvectors of systems 3.1.7 and 3.1.8 to be directly comparable in accordance with the rule defined in Section 5.3.3, it is necessary for the first and second columns of eigenvectors for system 3.1.7 to be interchanged.

Therefore matrix **E** is determined from $\mathbf{E} = \mathbf{S}^T$ as follows:

$$E = \begin{pmatrix} 0.707 & 0 & -0.707 \\ 0.5 & 0.707 & 0.5 \\ 0.5 & -0.707 & 0.5 \end{pmatrix}$$

and hence:

$$E^{-1} = \begin{pmatrix} 0.707 & 0.5 & 0.5 \\ 0 & 0.707 & -0.707 \\ -0.707 & 0.5 & 0.5 \end{pmatrix}$$

Additionally:

$$e'_x = \begin{bmatrix} S'_{0,0} \\ S'_{0,1} \\ S'_{0,2} \end{bmatrix} \quad e'_y = \begin{bmatrix} S'_{1,0} \\ S'_{1,1} \\ S'_{1,2} \end{bmatrix} \quad e'_z = \begin{bmatrix} S'_{2,0} \\ S'_{2,1} \\ S'_{2,2} \end{bmatrix}$$

Therefore:

$$e'_x = \begin{bmatrix} -0.8 \\ 0.424 \\ 0.424 \end{bmatrix} \quad e'_y = \begin{bmatrix} 0 \\ 0.707 \\ -0.707 \end{bmatrix} \quad e'_z = \begin{bmatrix} 0.6 \\ 0.566 \\ 0.566 \end{bmatrix}$$

Therefore, the values for $\mathbf{r}_1$, $\mathbf{r}_2$, $\mathbf{r}_3$ can be obtained:

$$r_1 := E^{-1} e'_x \quad r_2 := E^{-1} e'_y \quad r_3 := E^{-1} e'_z$$

$$r_1 = \begin{pmatrix} -0.142 \\ 0 \\ 0.99 \end{pmatrix} \quad r_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad r_3 = \begin{pmatrix} 0.99 \\ 0 \\ 0.142 \end{pmatrix}$$

Hence the matrix **R** can be established:

$$R := \begin{bmatrix} (r_1)_{0,0} & (r_1)_{1,0} & (r_1)_{2,0} \\ (r_2)_{0,0} & (r_2)_{1,0} & (r_2)_{2,0} \\ (r_3)_{0,0} & (r_3)_{1,0} & (r_3)_{2,0} \end{bmatrix} \quad R = \begin{pmatrix} -0.142 & 0 & 0.99 \\ 0 & 1 & 0 \\ 0.99 & 0 & 0.142 \end{pmatrix}$$

From which the value of det(**R**) can be found to be det(**R**) = – 1, that is to say that this particular degradation involves a reflection of the triad of eigenvectors.

The above process is repeated for each of the thirty-six cases in Figures 5.8, 5.9 and 5.10, and the established rotations and reflections are noted in the figures, and in Table 5.3. The data is not included in this thesis for reasons of space.

Additional to the foregoing method, it is also possible to derive the rotation axis and rotation angles, or the position of the mirror plane using matrices involving direction cosines, but this is not undertaken here because of time constraints.

5.3.3 The Significance of Eigenvalue Sequence

Mathcad presents the eigenvectors of a system as a matrix of column vectors with the columns in the same sequence as that in which the eigenvalues are calculated by the Mathcad algorithm [34]. Inspection of the eigenvalues contained in Appendix D shows that for the systems analysed in this section, the Mathcad algorithm presents the eigenvalues and eigenvectors in a sequence which ostensibly appears to vary from system to system. As an example, it can be seen that for many of the systems there is one eigenvalue with a value of 6, one eigenvalue close to 1, and another eigenvalue near the range 10-12. It would seem reasonable to suppose that during system

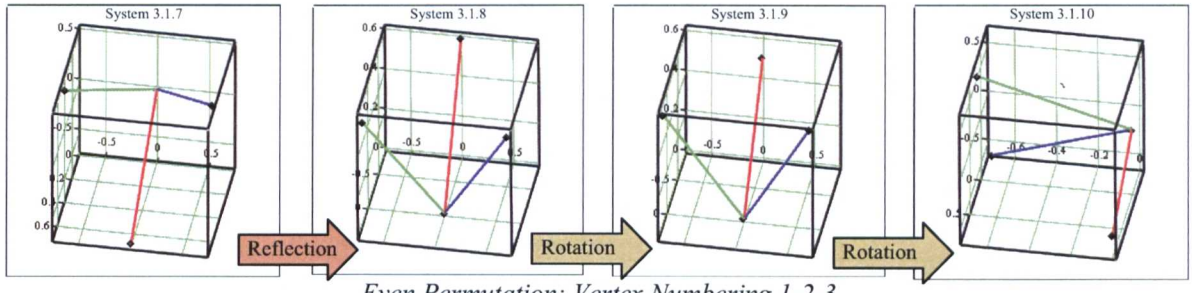
degradation through a sequence of fault states, these eigenvalues follow a logical evolution, so that the order in which to present the eigenvectors would be in a fixed sequence determined by the eigenvalues. This is the approach adopted in Figures 5.8, 5.9 and 5.10 as follows:

- Red eigenvector corresponds to eigenvalue = 6
- Blue eigenvector corresponds to eigenvalue ≈ 10-12
- Green eigenvector corresponds to eigenvalue ≈ 1

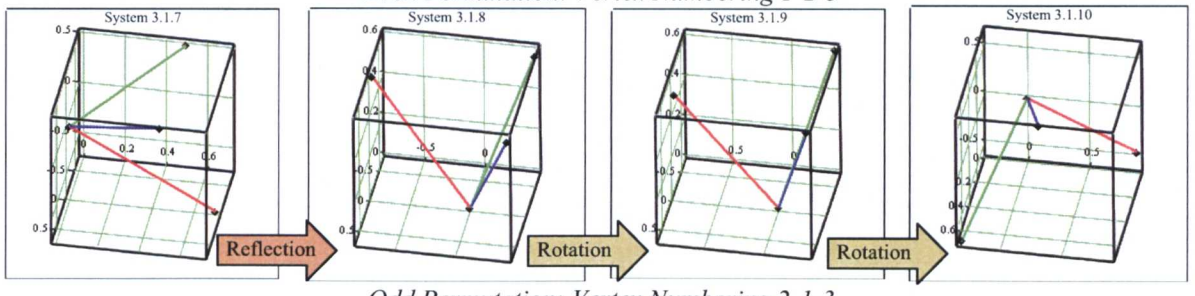
Nonetheless, it is quite possible that a more complex relationship exists that is not apparent at a cursory examination. However, the foregoing analysis of rotations and reflections provides an insight into this issue, since it is known that the matrix **R** must produce a value of ± 1, so that any result not producing such a value is assumed to be spurious. During the course of the detailed analysis of the thirty-six cases summarised here, it was shown that it is necessary to have the eigenvectors in a specific sequence in order for expression (5.1) to comply with the requirement for det(**R**) = +1 or det(**R**) = -1, and so the choice of eigenvector sequence adopted is considered justified.

≈

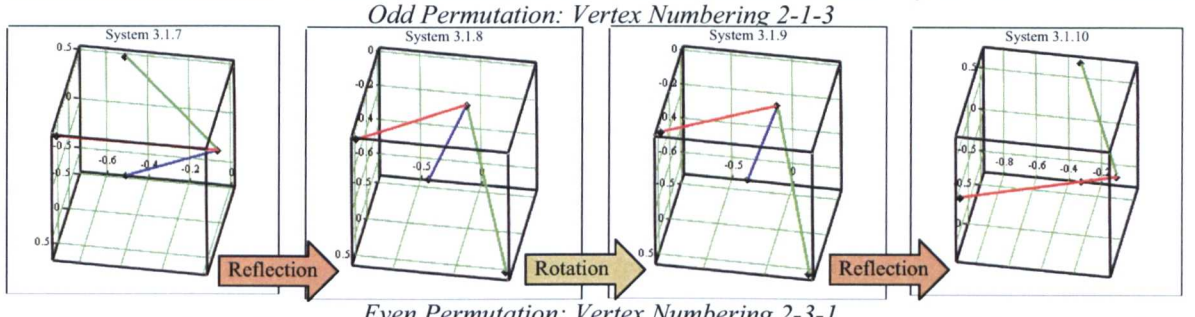
Further implications of these figures are examined following Figure 5.10.



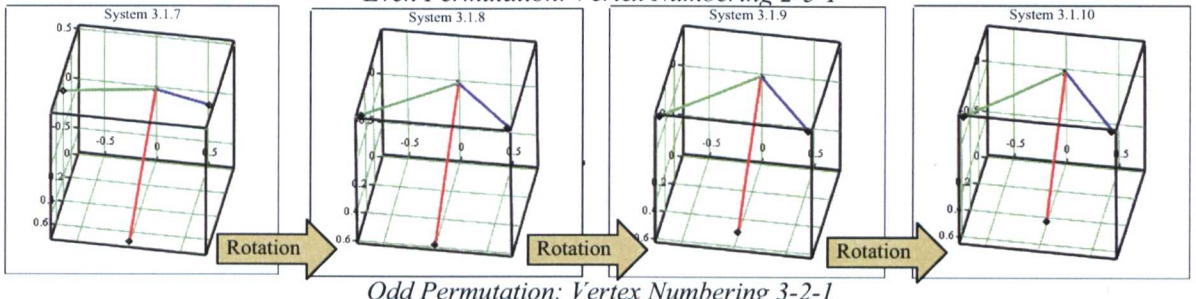
Even Permutation: Vertex Numbering 1-2-3



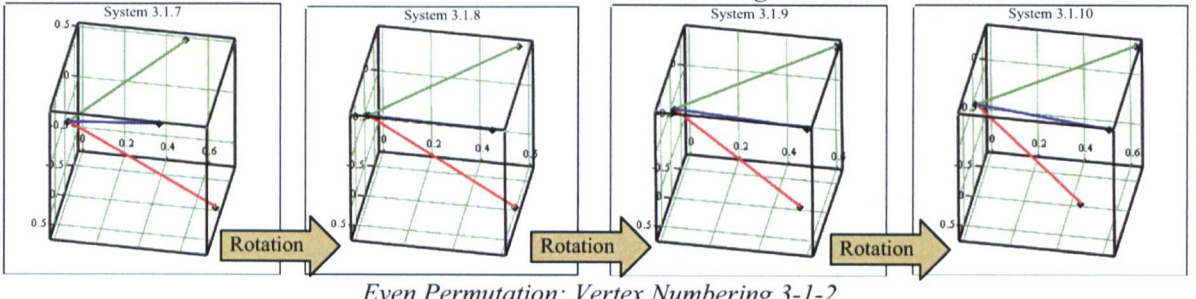
Odd Permutation: Vertex Numbering 2-1-3



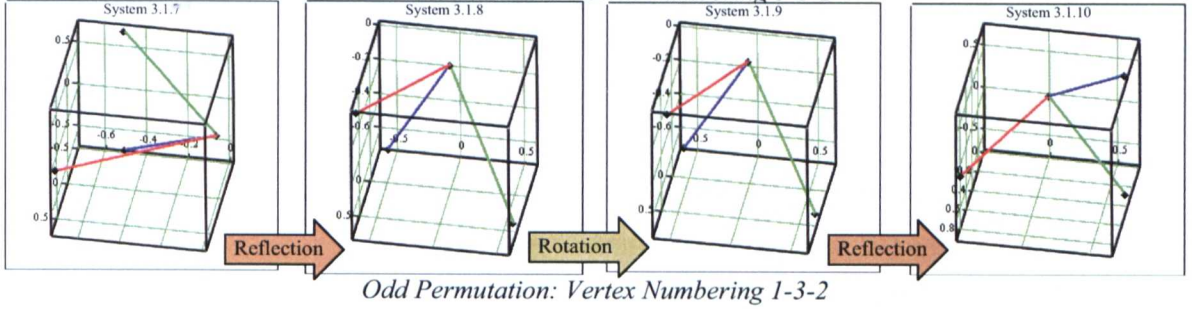
Even Permutation: Vertex Numbering 2-3-1



Odd Permutation: Vertex Numbering 3-2-1

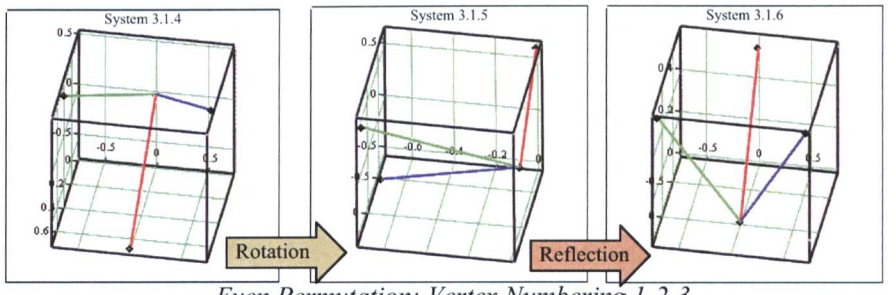


Even Permutation: Vertex Numbering 3-1-2

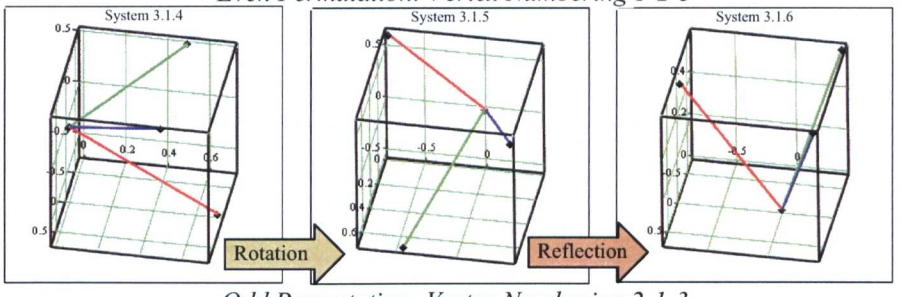


Odd Permutation: Vertex Numbering 1-3-2

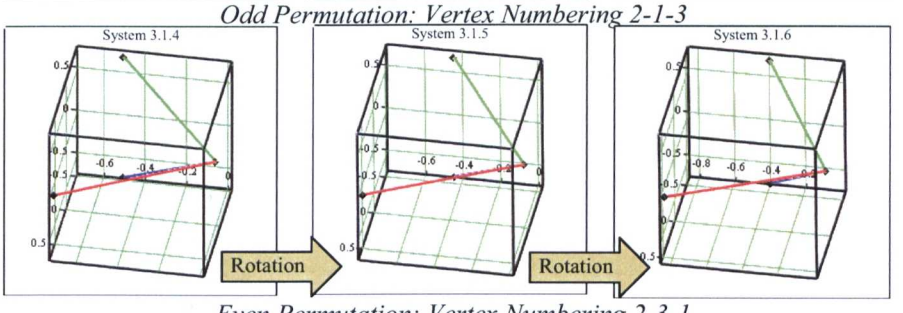
Figure 5.8: Variation in Eigenvectors with Even and Odd Permutations of Vertex Numbering for Degradation through a Sequence of Faults of 3-Bar Open Systems with Spherical or Planar Joints



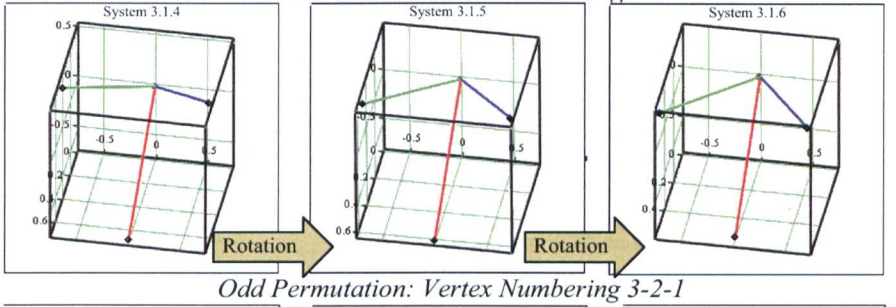
Even Permutation: Vertex Numbering 1-2-3



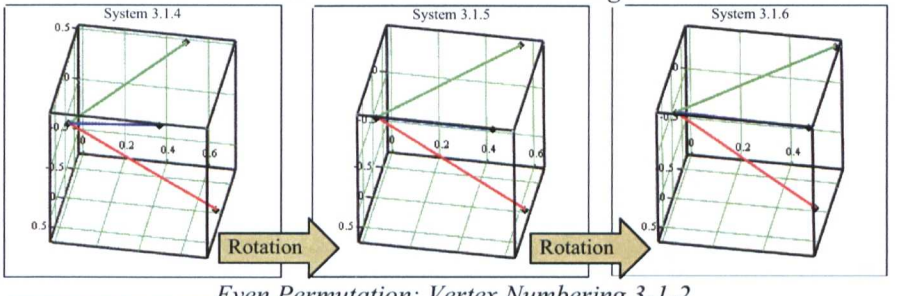
Odd Permutation: Vertex Numbering 2-1-3



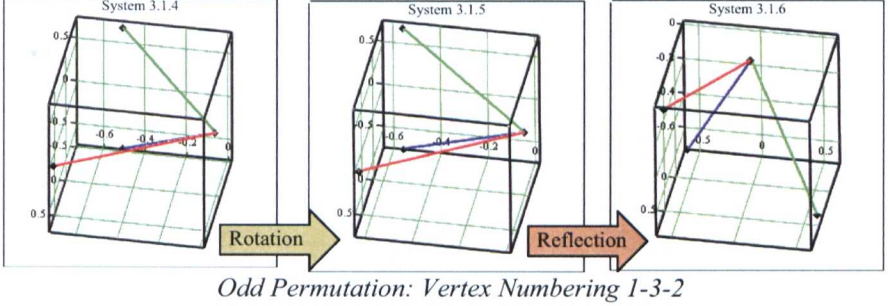
Even Permutation: Vertex Numbering 2-3-1



Odd Permutation: Vertex Numbering 3-2-1

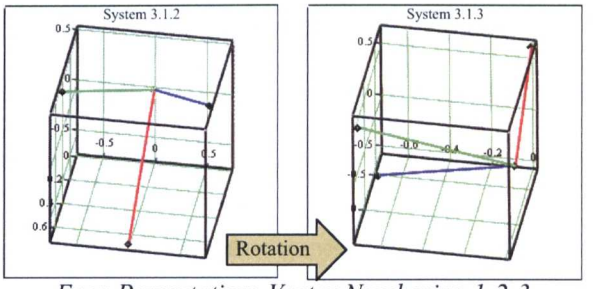


Even Permutation: Vertex Numbering 3-1-2

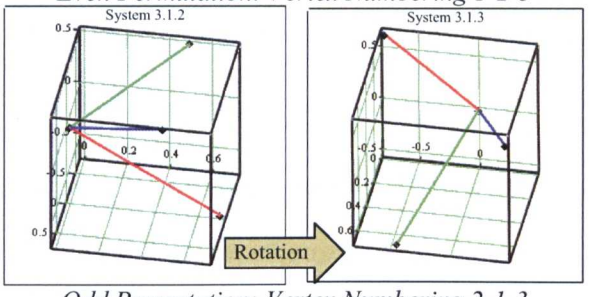


Odd Permutation: Vertex Numbering 1-3-2

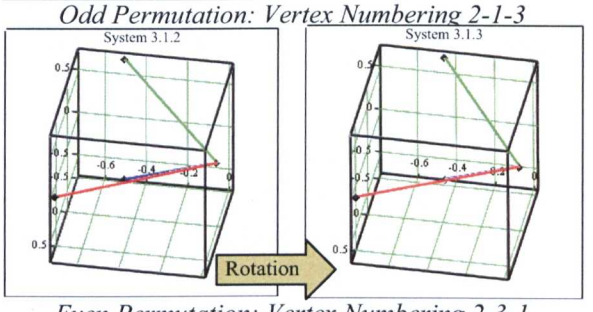
Figure 5.9: Variation in Eigenvectors with Even and Odd Permutations of Vertex Numbering for Degradation through a Sequence of Faults of 3-Bar Open Systems with Cylindric Joints



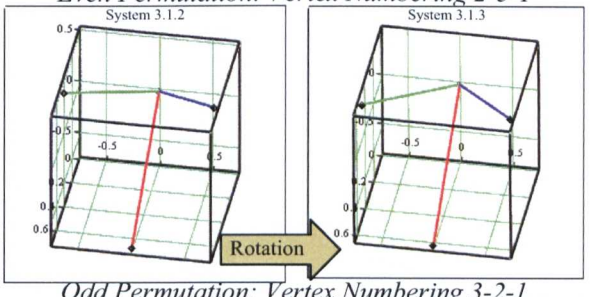
Even Permutation: Vertex Numbering 1-2-3



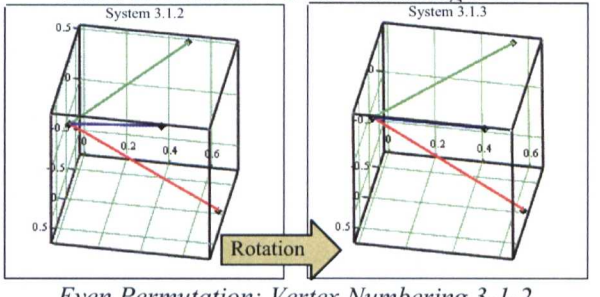
Odd Permutation: Vertex Numbering 2-1-3



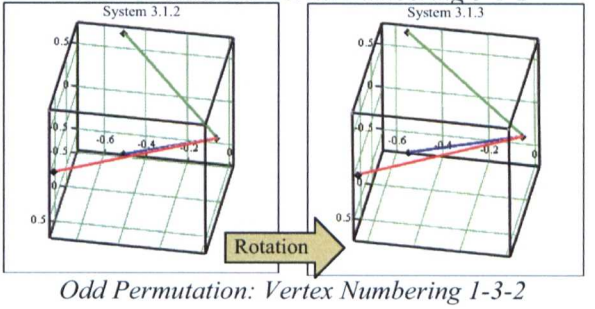
Even Permutation: Vertex Numbering 2-3-1



Odd Permutation: Vertex Numbering 3-2-1



Even Permutation: Vertex Numbering 3-1-2



Odd Permutation: Vertex Numbering 1-3-2

Figure 5.10: Variation in Eigenvectors with Even and Odd Permutations of Vertex Numbering for Degradation through a Sequence of Faults of 3-Bar Open Systems with Revolute, Prismatic or Screw Joints

The eigenvector behaviours illustrated in the foregoing three figures can be summarised as shown in Table 5.3. Additionally, the determinant of the Constraints Matrix corresponding to each of the systems is expanded symbolically, and these results are also included, showing that:

- the symbolic determinants contain the same terms throughout any one fault path
- there are three forms of the symbolic determinant established for the permutations in any one fault path
- each of the three forms of the expanded symbolic determinants identified occur in all fault paths for the systems selected, regardless of joint type.

Permutation	**System Number**				**Expanded Symbolic Determinant**
Planar or Spherical Joints (three dof)					
	3.1.7	3.1.8	3.1.9	3.1.10	
1-2-3	(Nominal)	Reflection	Rotation	Rotation	$(d_{11})(d_{22})(d_{33}) - (d_{11})(d_{23}d_{32}) - (d_{33})(d_{12}d_{21})$
2-1-3	(Nominal)	Reflection	Rotation	Rotation	$(d_{11})(d_{22})(d_{33}) - (d_{22})(d_{13}d_{31}) - (d_{33})(d_{12}d_{21})$
2-3-1	(Nominal)	Reflection	Rotation	Reflection	$(d_{11})(d_{22})(d_{33}) - (d_{11})(d_{23}d_{32}) - (d_{22})(d_{13}d_{31})$
3-2-1	(Nominal)	Rotation	Rotation	Rotation	$(d_{11})(d_{22})(d_{33}) - (d_{11})(d_{23}d_{32}) - (d_{33})(d_{12}d_{21})$
3-1-2	(Nominal)	Rotation	Rotation	Rotation	$(d_{11})(d_{22})(d_{33}) - (d_{22})(d_{13}d_{31}) - (d_{33})(d_{12}d_{21})$
1-3-2	(Nominal)	Reflection	Rotation	Reflection	$(d_{11})(d_{22})(d_{33}) - (d_{11})(d_{23}d_{32}) - (d_{22})(d_{13}d_{31})$
Cylindric Joints (two dof)					
	3.1.4	3.1.5	3.1.6		
1-2-3	(Nominal)	Rotation	Reflection		$(d_{11})(d_{22})(d_{33}) - (d_{11})(d_{23}d_{32}) - (d_{33})(d_{12}d_{21})$
2-1-3	(Nominal)	Rotation	Reflection		$(d_{11})(d_{22})(d_{33}) - (d_{22})(d_{13}d_{31}) - (d_{33})(d_{12}d_{21})$
2-3-1	(Nominal)	Rotation	Rotation		$(d_{11})(d_{22})(d_{33}) - (d_{11})(d_{23}d_{32}) - (d_{22})(d_{13}d_{31})$
3-2-1	(Nominal)	Rotation	Rotation		$(d_{11})(d_{22})(d_{33}) - (d_{11})(d_{23}d_{32}) - (d_{33})(d_{12}d_{21})$
3-1-2	(Nominal)	Rotation	Rotation		$(d_{11})(d_{22})(d_{33}) - (d_{22})(d_{13}d_{31}) - (d_{33})(d_{12}d_{21})$
1-3-2	(Nominal)	Rotation	Reflection		$(d_{11})(d_{22})(d_{33}) - (d_{11})(d_{23}d_{32}) - (d_{22})(d_{13}d_{31})$
Revolute, Prismatic or Screw Joints (one dof)					
	3.1.2	3.1.3			
1-2-3	(Nominal)	Rotation			$(d_{11})(d_{22})(d_{33}) - (d_{11})(d_{23}d_{32}) - (d_{33})(d_{12}d_{21})$

2-1-3	(Nominal)	Rotation			$(d_{11})(d_{22})(d_{33}) - (d_{22})(d_{13}d_{31}) - (d_{33})(d_{12}d_{21})$
2-3-1	(Nominal)	Rotation			$(d_{11})(d_{22})(d_{33}) - (d_{11})(d_{23}d_{32}) - (d_{22})(d_{13}d_{31})$
3-2-1	(Nominal)	Rotation			$(d_{11})(d_{22})(d_{33}) - (d_{11})(d_{23}d_{32}) - (d_{33})(d_{12}d_{21})$
3-1-2	(Nominal)	Rotation			$(d_{11})(d_{22})(d_{33}) - (d_{22})(d_{13}d_{31}) - (d_{33})(d_{12}d_{21})$
1-3-2	(Nominal)	Rotation			$(d_{11})(d_{22})(d_{33}) - (d_{11})(d_{23}d_{32}) - (d_{22})(d_{13}d_{31})$

Table 5.3: Eigenvector Behaviour with loss of Degrees of Freedom

It is useful to consider how the three symbolic determinant forms can give rise to the different eigenvector orientations, given the limited number of symbolic forms of determinant. The **form of the symbolic expansion of these determinants** is driven by the location of the zero elements in the constraints matrices – that is, by aspects of the connectivity of the graph. **The numerical values** are driven by the number of degrees of freedom existing between the vertices.

Taking this latter point further, close inspection of the derivation of the numerical values from the symbolic forms of the determinants shows the relationships presented in Table 5.4. When evaluating any of the symbolic determinants numerically, one of the terms will take its value from the unfailed joint (shown in green below), and one of the terms will take its value from the failed joint (shown in orange below). The combined values of these terms will control the numerical value of the determinant.

Permutation	**Symbolic Determinant**	**Determinant Term Source**		
		$(d_{12}d_{21})$	$(d_{13}d_{31})$	$(d_{23}d_{32})$
Planar or Spherical Joints (three dof)				
1-2-3	$(d_{11})(d_{22})(d_{33}) - (d_{11})(d_{23}d_{32}) - (d_{33})(d_{12}d_{21})$	Unfailed joint		Faulty joint
2-1-3	$(d_{11})(d_{22})(d_{33}) - (d_{22})(d_{13}d_{31}) - (d_{33})(d_{12}d_{21})$	Unfailed joint	Faulty joint	
2-3-1	$(d_{11})(d_{22})(d_{33}) - (d_{11})(d_{23}d_{32}) - (d_{22})(d_{13}d_{31})$		Faulty joint	Unfailed joint
3-2-1	$(d_{11})(d_{22})(d_{33}) - (d_{11})(d_{23}d_{32}) - (d_{33})(d_{12}d_{21})$	Faulty joint		Unfailed joint
3-1-2	$(d_{11})(d_{22})(d_{33}) - (d_{22})(d_{13}d_{31}) - (d_{33})(d_{12}d_{21})$	Faulty joint	Unfailed joint	
1-3-2	$(d_{11})(d_{22})(d_{33}) - (d_{11})(d_{23}d_{32}) - (d_{22})(d_{13}d_{31})$		Unfailed joint	Faulty joint
Cylindric Joints (two dof)				
1-2-3	$(d_{11})(d_{22})(d_{33}) - (d_{11})(d_{23}d_{32}) - (d_{33})(d_{12}d_{21})$	Unfailed joint		Faulty joint
2-1-3	$(d_{11})(d_{22})(d_{33}) - (d_{22})(d_{13}d_{31}) - (d_{33})(d_{12}d_{21})$	Unfailed joint	Faulty joint	

2-3-1	$(d_{11})(d_{22})(d_{33}) - (d_{11})(d_{23}d_{32}) - (d_{22})(d_{13}d_{31})$		Faulty joint	Unfailed joint
3-2-1	$(d_{11})(d_{22})(d_{33}) - (d_{11})(d_{23}d_{32}) - (d_{33})(d_{12}d_{21})$	Faulty joint		Unfailed joint
3-1-2	$(d_{11})(d_{22})(d_{33}) - (d_{22})(d_{13}d_{31}) - (d_{33})(d_{12}d_{21})$	Faulty joint	Unfailed joint	
1-3-2	$(d_{11})(d_{22})(d_{33}) - (d_{11})(d_{23}d_{32}) - (d_{22})(d_{13}d_{31})$		Unfailed joint	Faulty joint
Revolute, Prismatic or Screw Joints (one dof)				
1-2-3	$(d_{11})(d_{22})(d_{33}) - (d_{11})(d_{23}d_{32}) - (d_{33})(d_{12}d_{21})$	Unfailed joint		Faulty joint
2-1-3	$(d_{11})(d_{22})(d_{33}) - (d_{22})(d_{13}d_{31}) - (d_{33})(d_{12}d_{21})$	Unfailed joint	Faulty joint	
2-3-1	$(d_{11})(d_{22})(d_{33}) - (d_{11})(d_{23}d_{32}) - (d_{22})(d_{13}d_{31})$		Faulty joint	Unfailed joint
3-2-1	$(d_{11})(d_{22})(d_{33}) - (d_{11})(d_{23}d_{32}) - (d_{33})(d_{12}d_{21})$	Faulty joint		Unfailed joint
3-1-2	$(d_{11})(d_{22})(d_{33}) - (d_{22})(d_{13}d_{31}) - (d_{33})(d_{12}d_{21})$	Faulty joint	Unfailed joint	
1-3-2	$(d_{11})(d_{22})(d_{33}) - (d_{11})(d_{23}d_{32}) - (d_{22})(d_{13}d_{31})$		Unfailed joint	Faulty joint

Table 5.4: Sources of Terms from Symbolic Determinants

It can be seen that in each pair of determinants with identical symbolic forms, one of the pair may differ from the other in numerical value, because in one case a given term will represent, for example, an unfailed joint, and in the other of the pair, the same term will represent the faulty joint.

Symbolic forms of determinants will be shown to have considerable further significance, and will be discussed later in this thesis in the context of cospectral graphs, Chapter 9, and of 'Fault Classes', Chapter 10.

5.4 Eigenvalues as System Descriptors

It has now been shown that the systems in the dataset being investigated - Appendix C, Appendix D, and Appendix E - can be considered to be related by 'fault paths' which link groupings of systems together according to the configurations into which they degenerate under the action of faults. That such relationships exist in general within a more extensive population of systems is taken as a working hypothesis.

These 'fault path' relationships can be viewed as a 'graph of system graphs'. In such a graph, each edge represents the development of a system fault, and each vertex represents the interchange graph of the system in the particular fault state into which it has just developed along the edge leading to that vertex – Figure 5.11. It has already been shown that these progressive faults are mirrored in the behaviour of the system eigenvectors.

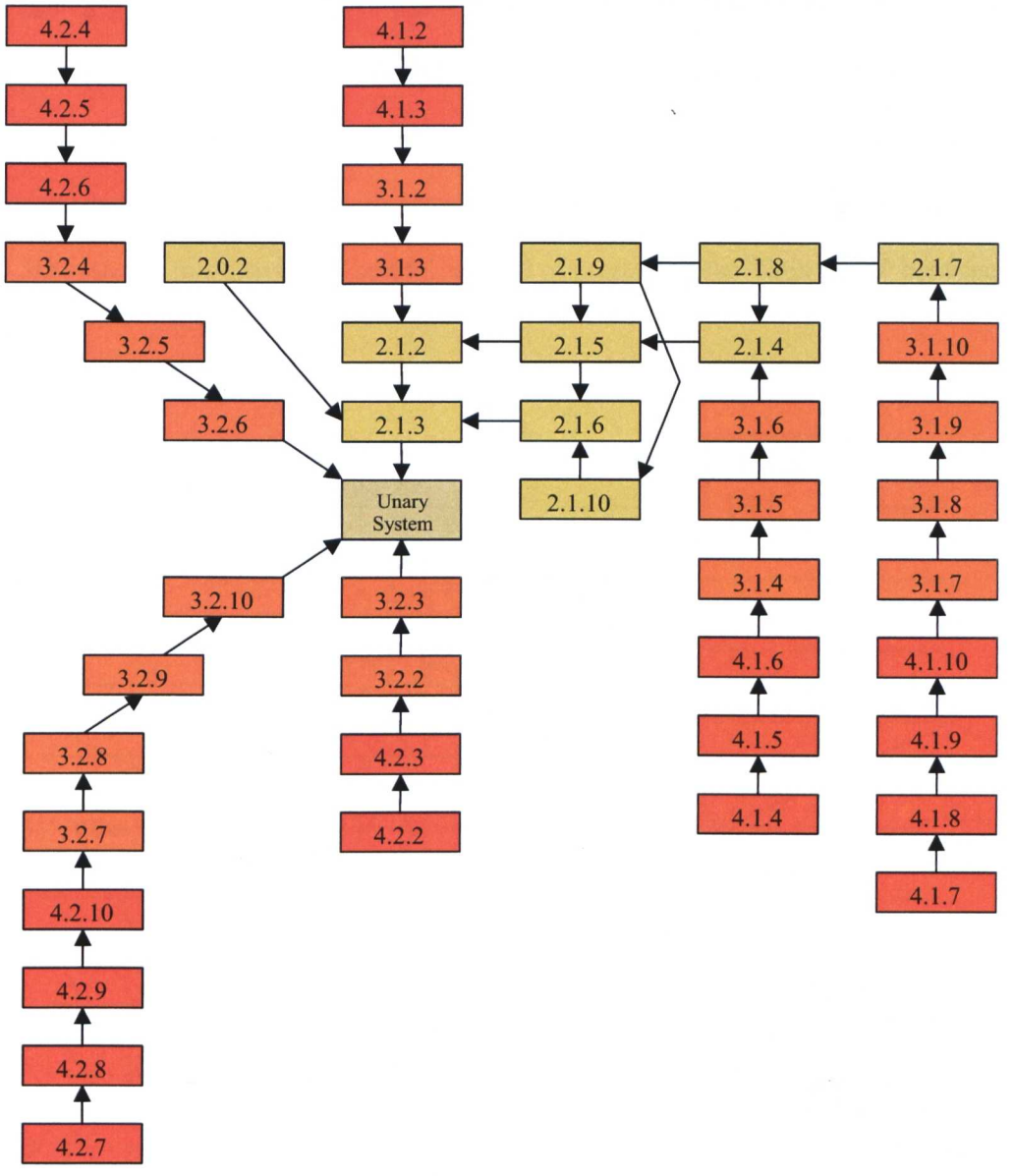


Figure 5.11:Graph of System Graphs

The concept of the 'graph of system graphs', illustrated in Figure 5.11, uses the systems in the referenced appendices, representing them solely by their reference numbers, although clearly, each reference number could be substituted by the relevant interchange graph. Thus, it may be considered that, at this stage, this representation exists within interchange graph space.

The foregoing parts of this chapter suggest that there is a supportable argument for regarding systems as collections of related points. Therefore, it is now considered how these arguments can be developed further, and how arrangements of points based on system eigenvalues might yield an improved presentation. Consider the isometric grid presented in Figure 5.12 – it can be seen that for 2D and 3D systems, this provides a helpful method of presenting the 'fault paths' between related systems.

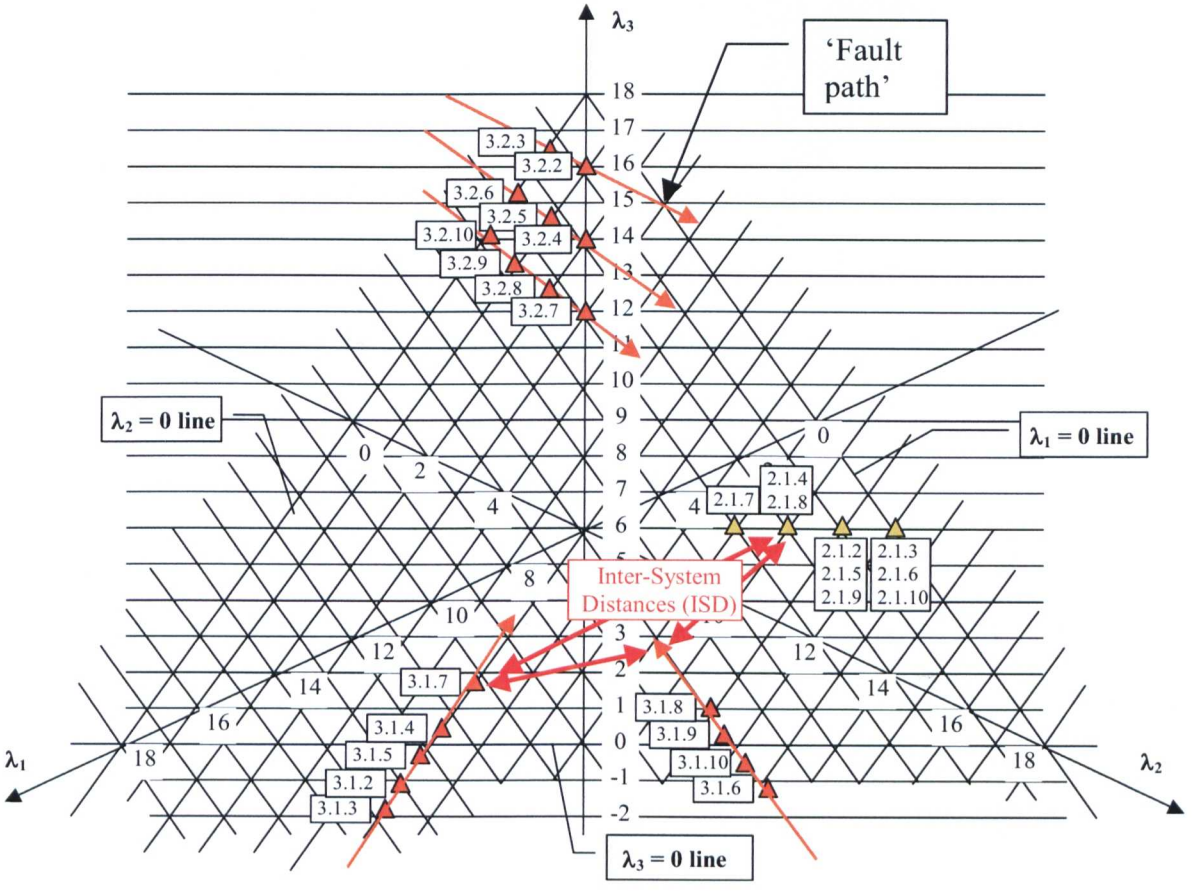


Figure 5.12: 2-Bar and 3-Bar Systems Represented on a Planar Grid

It will, however, be noted that in order to plot 2-Bar systems on this grid, it is necessary to parallel translate the $\lambda_3 = 0$ line to lie on the plane $\lambda_3 = 6$. It is not possible to plot the 2-Bar eigenvalues on the $\lambda_3 = 0$ plane, because, although all the points represented by (position) eigenvectors for 2-Bar systems lie on a single straight line, this line does not lie in the plane of the 3-Bar system eigenvalues. The reason for this is that the eigenvalues for any kinematic system will equal the trace of its Constraints Matrix, **C**, and this is 12 for 2-Bar systems and 18 for 3-Bar systems. This ensures that the line of 2-Bar systems and the plane of 3-Bar systems are not incident at any point – Figure 5.13.

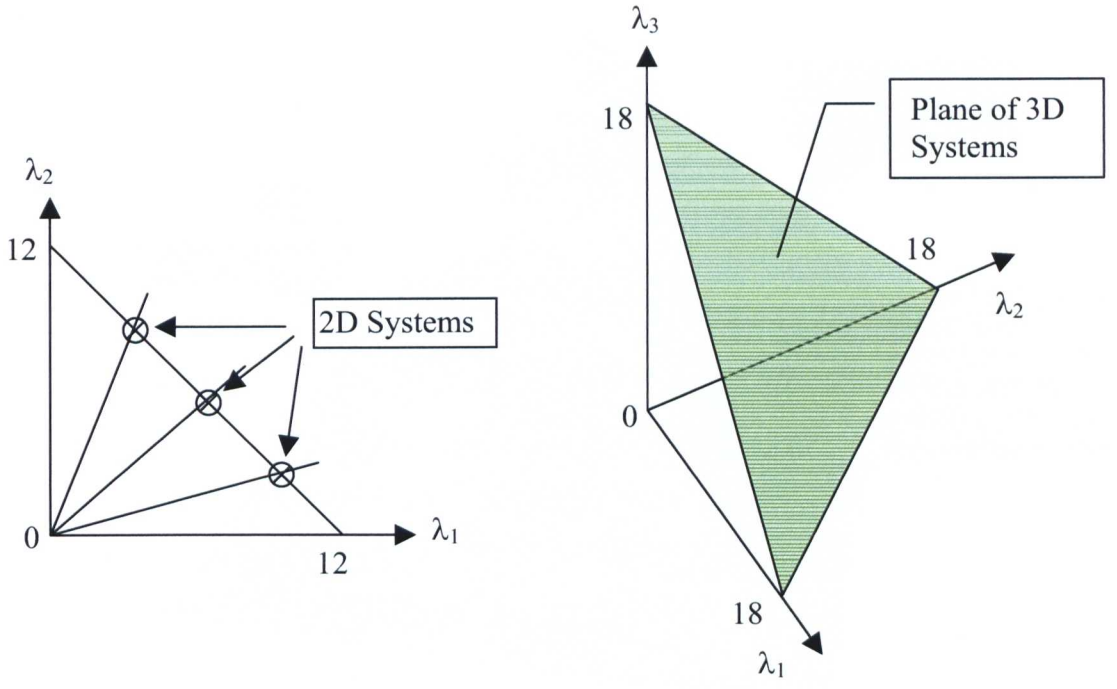


Figure 5.13: Line of 2-Bar Systems and Plane of 3D Systems

It can be shown that the eigenvalues for a 4-Bar system lie on a 3D hyperplane, and, in general, the eigenvalues of an n-Bar system will lie on an (n-1)-D hyperplane.

The degradation of a system under the action of progressive faults can, therefore, be visualised n-dimensionally with faults either causing the 'fault trajectory' to track on an (n-1)-D hyperplane (whose initial order is determined by the system(s) with the greatest number of vertices / links), or, at certain junctures, to jump from this hyperplane to one of the next lower dimension. The representations used in Figures 5.12 and 5.13 can be combined to create a visual representation of the n-space. Figure 5.14 attempts to show this for a four-dimensional space.

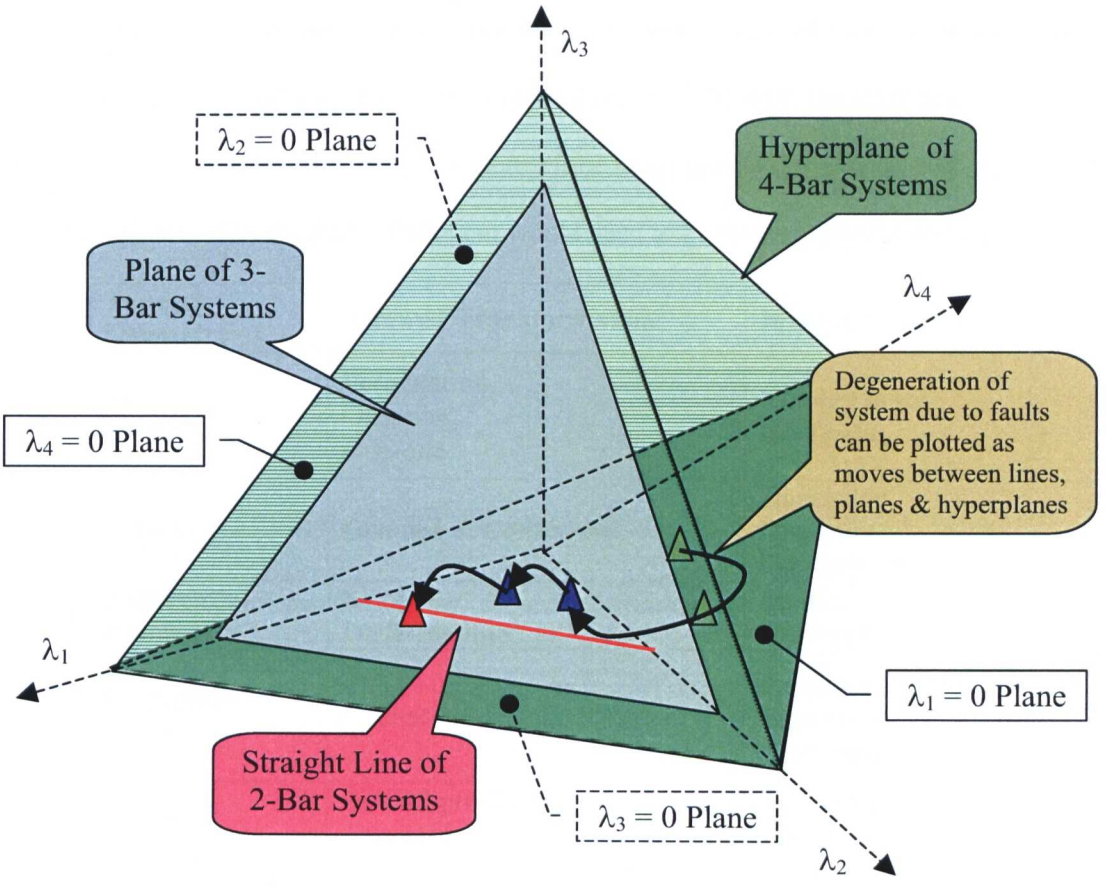


Figure 5.14: The Space of Systems visualised using System Eigenvalues as the Coordinates of Points in the Space

Thus, a means of visualising some aspects of the space containing the systems in which we are interested has been established, and this now allows further progress to be made. However, up to this point, it has been tacitly assumed that Euclidean space is satisfactory for dealing with the problem in hand. Euclidean space is often the space that tends to be assumed in the first instance, but it is important to remember that there are many other possibilities available. For this reason, some remarks regarding geometry are considered appropriate at this point.

5.5 Geometry of the Fault Path

5.5.1 Alternative Geometries

As stated above, all the arguments developed so far regarding the abstract space in which systems are represented have been based on the concepts of Euclidean geometry with familiar notions of distance, angle, shape and size. However, it should be remembered that Euclidean geometry is only one of a number of geometries available [55] – see Table 5.5 and Figure 5.15, which are extracted from / based on the referenced document. Table 5.5 presents a view of the Klein hierarchy of

geometries, modified by Rooney, which identifies the transformations and invariants associated with each of these more general geometries, and Figure 5.15 presents the inter-relationship of the geometries in schematic format. If one or more of the other geometries is used with, or instead of, Euclidean geometry, then the above treatment would need to be revised appropriately.

Type of Geometry	**Allowed Transformation**	**Invariants**
Set Theory	Permutation, Ordering, One-one	Number in set, Cardinality
Topology	One-one & Continuous	Dimension, One point, Connectivity
Differential Geometry	Smooth, Differentiable	Metric, Nearness
Projective Geometry	Central projection	Incidence, Separation, Cross ratio
Affine Geometry	Parallel projection, General Linear, Stretch, Shear, Boost	Infinite points, Parallelism, Between-ness
Euclidean Geometry	Orthogonal, Rotation, Translation, Screw, Reflection, Inversion	Distance, Angle, Shape, Size

Table 5.5: Transformations and Invariants of a Klein Hierarchy of Geometries, modified by Rooney [55]

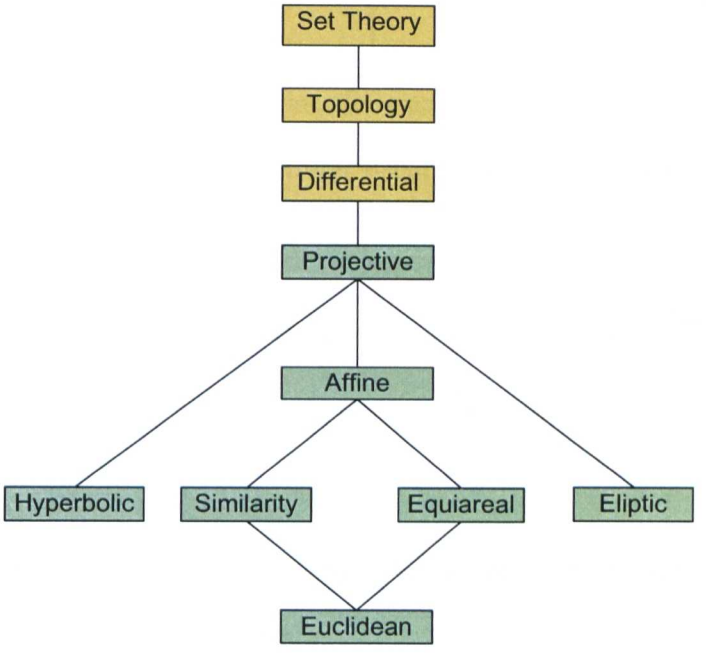


Figure 5.15: Structure of a Klein Hierarchy of Geometries, modified by Rooney [55]

Continuing in Euclidean geometry, it is useful at this point to consider the geometry of a 'fault path' in greater detail. A formal demonstration that 2-Bar systems will lie on a line, 3-Bar systems on a plane, and 4-Bar and larger systems on a hyperplane of the appropriate order is now provided.

5.5.2 Determinants and Geometry

It can be shown using principles based on projective geometry [58], that for three points in 2D space to lie on a straight line, their Cartesian coordinates must satisfy the determinant equation:

$$\begin{vmatrix} x_3 & y_3 & 1 \\ x_2 & y_2 & 1 \\ x_1 & y_1 & 1 \end{vmatrix} = 0 \qquad \text{.......................................} (5.2)$$

Similarly, the same proof can be extended to show that if four points in 3D space lie on the same plane, then:

$$\begin{vmatrix} x_4 & y_4 & z_4 & 1 \\ x_3 & y_3 & z_3 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_1 & y_1 & z_1 & 1 \end{vmatrix} = 0 \qquad \text{.......................................} (5.3)$$

This can be further extended to the following necessary condition in the general case of (n +1) points on a hyperplane in n-D space:

$$\begin{vmatrix} x_{n+1} & y_{n+1} & z_{n+1} & \cdots & w_{n+1} & 1 \\ x_n & y_n & z_n & \cdots & w_n & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_3 & y_3 & z_3 & \cdots & w_3 & 1 \\ x_2 & y_2 & z_2 & \cdots & w_2 & 1 \\ x_1 & y_1 & z_1 & \cdots & w_1 & 1 \end{vmatrix} = 0 \qquad \text{..........................} (5.4)$$

Using these expressions, it is possible to examine the eigenvectors obtained for the degraded systems lying on the example fault path(s) in greater detail. This is carried out in steps in the following sections. These steps are:

1. Verify that the above expressions yield the required zero determinant when applied to 2-Bar systems lying on a straight line, 3-Bar systems lying on a plane, and 4-Bar systems lying on a hyperplane.

2. Establish whether a combination of 2-Bar and 3-Bar systems that lie on the same fault path can be shown to lie on either a straight line or a plane.

3. If the previous test is passed, establish whether a combination of 2-Bar, 3-Bar and 4-Bar systems that lie on the same fault path can be shown to lie on either a straight line, a plane, or a hyperplane.

If the systems lying on the fault path can be shown to pass all three of these tests, then there is evidence that fault path systems have a linear geometric relationship with one another. If the tests are not passed, then it will have been shown that more sophisticated methods are needed to establish whether any other, less immediately apparent, geometric relationships exist. In this latter case, the need to employ non-Euclidean geometries may arise, but not necessarily.

5.5.2.1 2-Bar Systems Lying on a Line

Taking the eigenvalues derived from the Constraints Matrix, **C**, for a typical group of 2-Bar systems from Appendix C, and substituting in Equation 5.2, the determinant is easily evaluated to show that expression 5.5 is true:

$$\begin{vmatrix} 3 & 9 & 1 \\ 6 & 6 & 1 \\ 2 & 10 & 1 \end{vmatrix} = 0 \qquad \text{..... (5.5)}$$

... thus confirming that the points chosen lie on a straight line, as shown previously in Figure 5.13:

In order to ensure that no unforeseen issues arise for particular cases where two or more eigenvalues have equal value, and where zero eigenvalues are present, either separately, or in combination, two additional tests are run. The results show that a determinant value of zero is still arrived at:

$$\begin{vmatrix} 1 & 11 & 1 \\ 0 & 12 & 1 \\ 2 & 10 & 1 \end{vmatrix} = 0 \qquad \text{..... (5.5a)}$$

$$\begin{vmatrix} 3 & 9 & 1 \\ 6 & 6 & 1 \\ 0 & 12 & 1 \end{vmatrix} = 0 \qquad \text{..... (5.5b)}$$

For this comparatively trivial case, therefore, it can be shown that all the 2-Bar systems in Appendix C lie on a unique straight line in 2D space. Since the eigenvalues will always add up to the trace of the Constraints Matrix, **C**, ie 12 for 2D, it is clear that any 2-Bar system will lie on the straight line $x + y = 12$.

5.5.2.2 *3-Bar Systems Lying in a Plane*

The test applied to 2-Bar systems can be repeated for 3-Bar systems by substituting values for 3-Bar systems taken from Appendix D into expression 5.3. Choosing four typical 3D points on a plane:

$$\begin{vmatrix} 13.071 & 6 & -1.071 & 1 \\ 13.81 & 6 & -1.81 & 1 \\ 11.657 & 6 & 0.343 & 1 \\ 12.403 & 6 & -0.403 & 1 \end{vmatrix} = 0 \qquad \text{..... (5.6)}$$

However, it will be noted that the above systems constitute a special case since column two is a multiple of column four, and hence the equation is identically satisfied. Nonetheless, this is a valid result, since it represents the particular case where the points lie on a plane positioned at y = 6. The test is repeated, but introducing points representing different 3D (closed) systems from the dataset in Appendix D to disrupt the linear dependence of columns in the previous example. In the two cases shown below, the determinant value is still zero, as required, and the points are proved to lie on a unique plane:

$$\begin{vmatrix} 13.071 & 6 & -1.071 & 1 \\ 13.81 & 6 & -1.81 & 1 \\ 11.657 & 6 & 0.343 & 1 \\ 1.319 & 0 & 16.681 & 1 \end{vmatrix} = 0 \qquad \text{..... (5.6a)}$$

$$\begin{vmatrix} 13.071 & 6 & -1.071 & 1 \\ 13.81 & 6 & -1.81 & 1 \\ 3.31 & 2 & 12.69 & 1 \\ 1.319 & 0 & 16.681 & 1 \end{vmatrix} = 0 \qquad \text{..... (5.6b)}$$

Thus, in the same way that it was proved that all the 2-Bar systems in Appendix C lie on a unique straight line in 2D space, it can now be shown that all the 3-Bar systems in Appendix D lie on a unique plane in 3D space. Similarly, since the eigenvalues will always add up to the trace of the Constraints Matrix, **C**, ie 18 for 3-Bar systems, any 3-Bar system in the dataset will lie on the plane $x + y + z = 18$.

5.5.2.3 *4-Bar Systems lying on a Hyperplane*

The test can now be repeated using the 4D version of Equation 5.4 to show that four-bar systems taken from the dataset all lie on a hyperplane in 4D space:

$$\begin{vmatrix} 9.09 & 14.09 & 2.91 & -2.09 & 1 \\ 6 & 6 & 14.66 & -2.66 & 1 \\ -1.406 & 1 & 7.556 & 16.85 & 1 \\ 8.472 & 12.472 & 3.528 & -0.472 & 1 \\ 8.217 & 13.217 & 3.783 & -1.217 & 1 \end{vmatrix} = 0 \quad \text{..... (5.7)}$$

Thus the test is passed satisfactorily, but in order to test that the zero eigenvalue situation does not cause unforeseen difficulties, substitution of two further 4-Bar systems from the dataset is carried out. This shows that a determinant value of zero is still achieved.:

$$\begin{vmatrix} 9.09 & 14.09 & 2.91 & -2.09 & 1 \\ 6 & 6 & 14.66 & -2.66 & 1 \\ -1.406 & 1 & 7.556 & 16.85 & 1 \\ 0 & -1.234 & 7.816 & 17.418 & 1 \\ 0 & -1.44 & 6 & 19.44 & 1 \end{vmatrix} = 0 \quad \text{..... (5.7a)}$$

Once again, since the eigenvalues will always add up to the trace of the Constraints Matrix, **C**, ie 24 for 4-Bar systems, any 4-Bar system from the dataset will lie on the hyperplane $x + y + z + w = 24$.

Thus, it has now been confirmed that the 2-Bar, 3-Bar, and 4-Bar systems in the dataset follow the expected trends – that is to say that test (1) identified earlier in 5.5.2 is satisfied. It is therefore now possible to proceed to investigate whether there are more interesting relationships, by applying tests (2) and (3).

5.5.3 The Nature of Multi-Dimensional Fault Paths

It has been shown that all 2-Bar systems lie on a unique straight line, all 3-Bar systems lie on a unique plane, and all 4-Bar systems lie on a unique hyperplane. However, it is of interest to establish whether a fault path which includes systems from a degradation sequence and hence from all these categories can be shown to lie on some particular type of locus. If this can be shown, then this is useful information about the nature of the fault path, but cannot be used to draw inferences about the applicability of any one geometry. If it is not possible to show that the fault path lies on any particular locus, then a more in-depth investigation into Euclidean and other geometries (Table 5.5, Figure 5.15) will in any case be needed, and this will require further investigation.

Consider the plane defined by the determinant in Equation 5.6, and run two tests replacing one of the 3-Bar systems with one of two 2-Bar systems. Both points when substituted individually allow the determinant to pass the test:

$$\begin{vmatrix} 13.071 & 6 & -1.071 & 1 \\ 13.81 & 6 & -1.81 & 1 \\ 11.657 & 6 & 0.343 & 1 \\ 1 & 11 & 0 & 1 \end{vmatrix} = 0 \quad \text{..... (5.8)}$$

$$\begin{vmatrix} 13.071 & 6 & -1.071 & 1 \\ 13.81 & 6 & -1.81 & 1 \\ 11.657 & 6 & 0.343 & 1 \\ 2 & 10 & 0 & 1 \end{vmatrix} = 0 \quad \text{..... (5.9)}$$

However, if expressions 5.8 and 5.9 define the same plane, then it should be possible to pass the test with both 2-Bar systems substituted simultaneously, as in Equation 5.10. However, evaluation shows that this is not so. Clearly the two planes implied by 5.8 and 5.9 are not the same.

$$\begin{vmatrix} 13.071 & 6 & -1.071 & 1 \\ 13.81 & 6 & -1.81 & 1 \\ 2 & 10 & 0 & 1 \\ 1 & 11 & 0 & 1 \end{vmatrix} = 4.43 \quad \text{..... (5.10)}$$

Closer inspection shows that equations 5.8 and 5.9 are special cases where the three 3D points (3-Bar systems) all lie on a straight line in the x-z plane, positioned at y = 6. In both tests, it is possible for planes satisfying the relationship to be identified, defined by the three 3D points lying on the straight line forming the base of a triangle, and the 2D point forming the apex of the triangle. If both the planes in 5.8 and 5.9 were coincident, then 5.10 would have a zero result. This is not the case, so in this situation the planes can be seen to be non-coincident.

This point can be further demonstrated by taking Equation 5.9, and substituting a 3D point (3-Bar system) which will not form this three-point baseline:

$$\begin{vmatrix} 13.071 & 6 & -1.071 & 1 \\ 13.81 & 6 & -1.81 & 1 \\ 3.31 & 2 & 12.69 & 1 \\ 2 & 10 & 0 & 1 \end{vmatrix} = -17.73 \quad \text{..... (5.9a)}$$

Again, this produces a non-zero result which shows that the planes identified in 5.8 and 5.9 are simply a result of the particular formation of points present in those equations, and have no additional significance.

Now, the outcome of the investigation begins to become clear. However, as a further test, consider the following fault path (Figure 5.16) which shows the degradation of a 4-bar system through 3-bar into 2-bar, using the 'vertex identification with loss of edge' approach. (Note that it is considered reasonable to 'pad' with zeroes the 'missing' dimensions, where necessary):

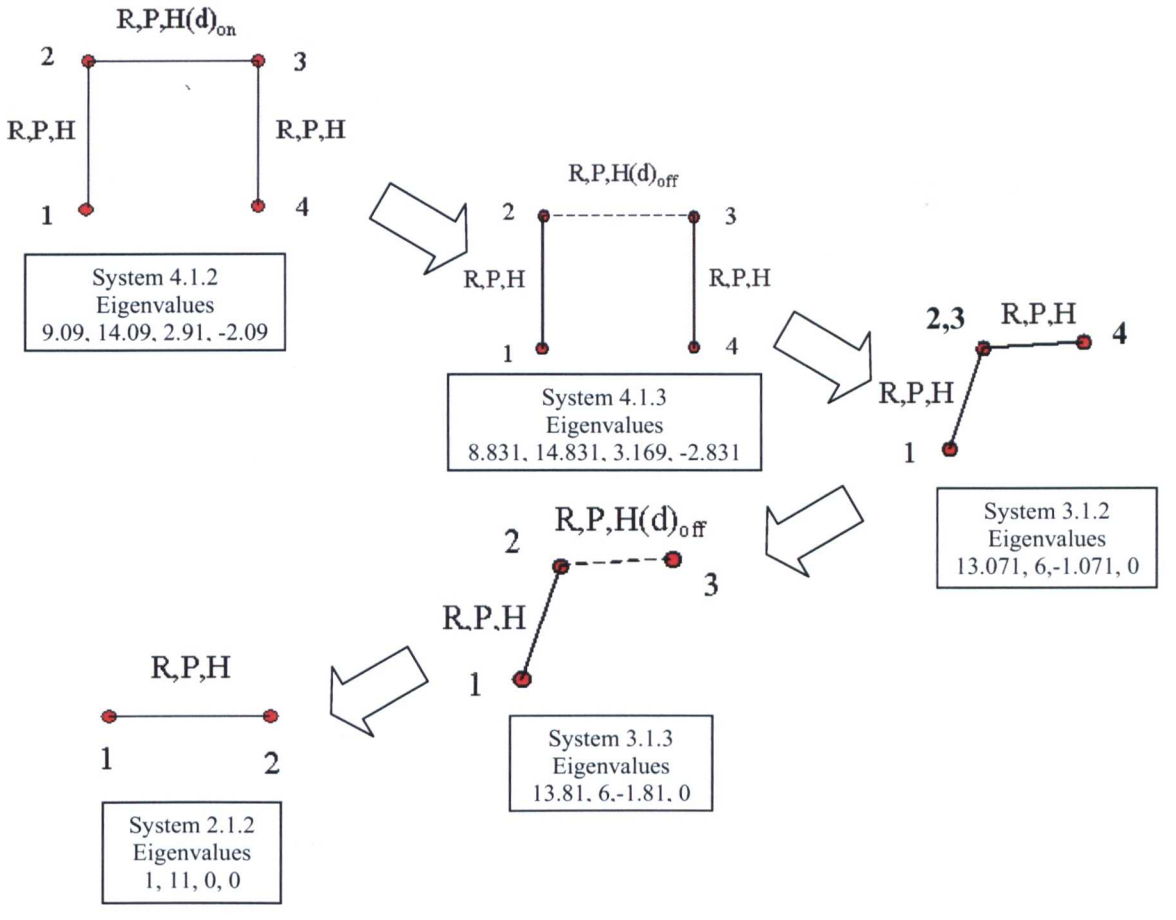


Figure 5.16: Example of an n-Dimensional Fault Path (n= 4)

(Eigenvalues for each system as indicated)

The five points lying on the fault path shown in Figure 5.16 are substituted into the 4D version of Equation 5.4. If it can be shown that all five points, (1 x 2D, 2 x 3D and 2 x 4D), when substituted into this determinant form have a determinant value of zero, then this would show that the locus of the points lies on a hyperplane.

$$\begin{vmatrix} 1 & 11 & 0 & 0 & 1 \\ 13.81 & 6 & -1.81 & 0 & 1 \\ 13.071 & 6 & -1.071 & 0 & 1 \\ 8.831 & 14.831 & 3.169 & -2.831 & 1 \\ 9.09 & 14.09 & 2.91 & -2.09 & 1 \end{vmatrix} = -36.14 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5.10)$$

The expression yields a non-zero result, so that the condition is not satisfied, and therefore it is proved that the points of a 4-dimensional fault path do not lie on a hyperplane.

Thus, there is not an immediately apparent, simple relationship that defines the locus of a fault path identifiable at this time. However, this result does not mean that there is not some other type of locus for describing n-dimensional fault paths – the points might just lie on some other, yet to be

determined, locus. Also, it should be understood that the theory being developed with regard to fault paths is not built upon the requirement that such a locus exists, simply that if this could be proved, then the richness of the representation developed would be enhanced. Further investigation into this subject is considered a useful topic for future work, and this is discussed further in Section 12.2.1.

Chapter 6
Inter-System Distance (ISD)

As shown in Chapter 5, it is fundamental to successful implementation of the proposed vision of the abstract Mechanical Design Space for one or more measures, or metrics, to be available for describing how similar (close together), or dissimilar (far apart) two systems are within the Design Space. This chapter examines how various established metrics can be used to meet this objective, and the influence that the choice of metric has on what is understood by the proximity of one kinematic system to another.

The concept of 'Inter-System Distance' (ISD) is central to the ability to draw comparisons between kinematic systems. The ISD may be associated with conceptual, functional, design principle, physical, or even philosophical characteristics, and later chapters of this thesis explore in greater detail how such a representation, based on kinematic characteristics, may be achieved in practice.

The topic is developed in three steps:

Section 6.1 reviews a selection of well known linear and quadratic metrics, and shows how the mathematical definitions of these result in different interpretations of the apparently simple concept of distance between two points.

Section 6.2 takes the metrics described in the previous section, and demonstrates the results that are achieved when each of them is used to measure the distance between two points. It is shown that choice of metric fundamentally affects how this 'inter-system distance' is interpreted.

Section 6.3 shows how the adoption of different metrics affects the results obtained for ISD by evaluating each of the metrics discussed for pairs of kinematic systems taken from the example database. The significance of this topic, which underlies much of the ISD evaluation undertaken in later chapters, is demonstrated, proving that it is important to recognise that any set of ISDs arrived

at can only be but one of many possible views of kinematic system relationships that can, potentially, be derived.

The following definition is adopted:

'**inter-system distance** – a specific instance of a metric, referring to a distance measure between two kinematic systems represented in an abstract space.'

6.1 A Selection of Metrics

Whilst many metrics may be constructed, a limited set of linear and quadratic measures are explored here. Some of these provide a pragmatic approach to constructing the necessary system comparisons. A more extensive review could prove to be a fruitful follow-on activity to this work – see Chapter 12.

6.1.1 Linear Metrics

One example of a linear metric is the 'taxicab' metric, where the distance between two points in the space is defined to be the signed sum of the coordinate differences of the points. For the case of two points in 2D space, this is:

$$\delta = \sum_{-1}^{+1} |x_2-x_1| + \sum_{-1}^{+1} |y_2-y_1| \quad \text{..... (6.1)}$$

where (x_1, y_1) and (x_2, y_2) are the coordinates of the two points in 2D space, and δ represents the taxicab distance between them. Note that this expression will yield 3^2 different permutations, depending on the sign ascribed to each of the moduli, since there are two dimensions.

For the general case of two points in n-dimensional space, the relationship becomes:

$$\delta = \sum_{-1}^{+1} |x_2-x_1| + \sum_{-1}^{+1} |y_2-y_1| \quad \text{.....} + \sum_{-1}^{+1} |w_2-w_1| \quad \text{..... (6.2)}$$

where $(x_1, y_1, \ldots w_1)$ and $(x_2, y_2, \ldots w_2)$ are the coordinates of the two points in n-dimensional space. Note that this relationship will yield 3^n different permutations, depending on the sign ascribed to each of the moduli (where n is the number of dimensions).

The nature of these distance metrics is made more apparent by choosing the 2D case with the moduli signs all taken as +ve. The result is shown for an ISD of $\delta = 4$. It can be seen that the effect

of the moduli within the expression is to produce diamond-shaped lines containing all points at equal distance, δ, from the origin (in Figure 6.1, δ = 4):

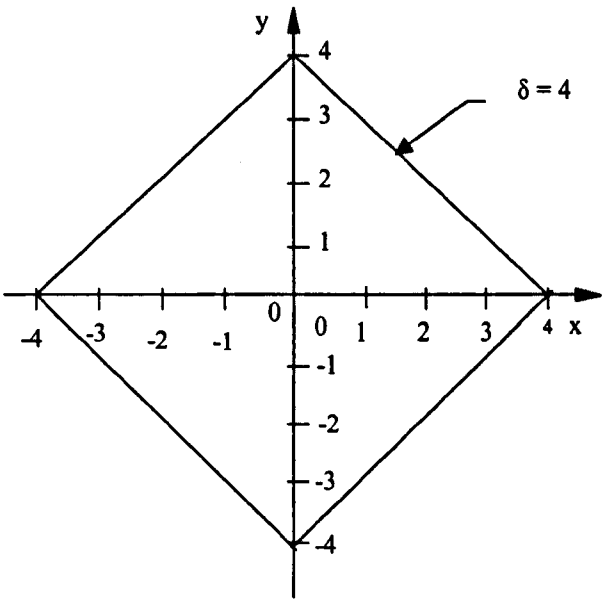


Figure 6.1: Lines of Distance δ = 4 from the Origin for a Linear (Taxicab) Metric

6.1.2 Quadratic Metrics

The general form of a quadratic metric for two points in a 2D space is given by the expression:

$$\delta = \sqrt{\sum_{-1}^{+1} 0\,(x_2-x_1)^2 \quad \sum_{-1}^{+1} 0\,(y_2-y_1)^2} \qquad \text{.....(6.3)}$$

where the notation has the same meaning as in the linear metric example.

The general form for an n-dimensional space becomes:

$$\delta = \sqrt{\sum_{-1}^{+1} 0\,(x_2-x_1)^2 \quad \sum_{-1}^{+1} 0\,(y_2-y_1)^2 + \quad \ldots\ldots \quad \sum_{-1}^{+1} 0\,(w_2-w_1)^2} \qquad \text{.....(6.4)}$$

There are 3^n different permutations, depending on the sign applied to each of the terms. Three examples of these expressions that give particular quadratic metrics are considered:

- **Elliptic (Euclidean)** (where all the signs in Equation (6.4) take the value +1).
- **Parabolic** (where the sign of at least one, but not all, of the terms in Equation (6.4) takes the value +1, and the remaining signs are 0).
- **Hyperbolic (Pseudo-Euclidean)** (where the sign of at least one, but not all, of the terms in Equation (6.4) takes the value +1, and the remaining signs are -1).

A brief explanation of these metrics in the context of complex, dual and double numbers is given by Rooney [51]. The metrics described above are discussed in the following paragraphs for the 2D case.

6.1.2.1 Elliptic (Euclidean) Metric in 2D Space

In the case of the Elliptic (Euclidean) metric, the distance, δ, between the two points is defined as the square root of the sum of the squares of the differences of the point coordinates – that is to say that in Equation (6.4), the signs are set to all be equal to +1.

This yields the 'standard' Pythagorean metric, where:

$$\delta = \{(x_2 - x_1)^2 + (y_2 - y_1)^2\}^{1/2} \quad \text{..... (6.5)}$$

and if points equidistant from the origin are plotted, a graph of the form shown in Figure 6.2 is obtained. This shows how, for this metric, the lines of equal distance from the origin are concentric circles:

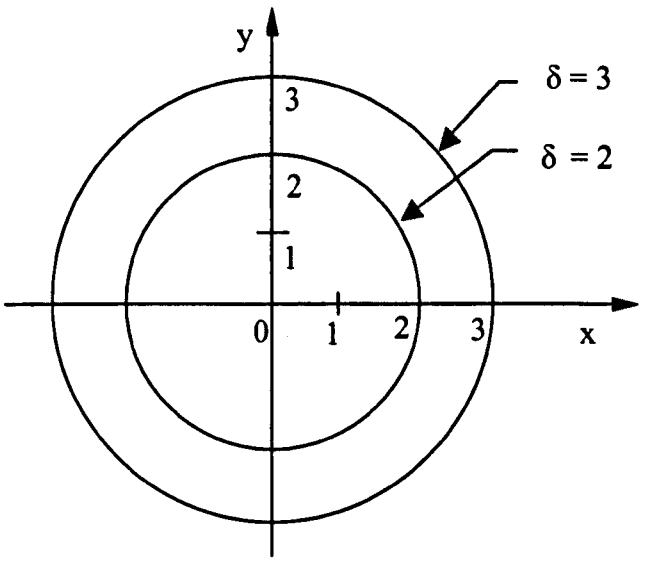


Figure 6.2: Lines of Equal Distance from the Origin for an Elliptic (Euclidean) Metric

6.1.2.2 Parabolic Metric in 2D Space

One example of this metric is obtained from Equation (6.4) by setting the sign of the leading term to +1, and all other signs to zero.

$$\delta = \{(x_2 - x_1)^2\}^{1/2} \quad \text{..... (6.6)}$$

When points of equal distance from the origin are plotted for the Parabolic case, examination of the relationship in Equation (6.6), shows that points of equal distance lie on straight lines parallel to the y axis, but also that for each distance, there are two lines, one on each side of the origin (see Figure 6.3). Note that Equation (6.6) could just as easily be written in terms of y rather than x, by setting the sign of the y term to +1, and other signs to zero. This would give rise to a different parabolic metric with the lines of equal distance from the origin being parallel to the x-axis.

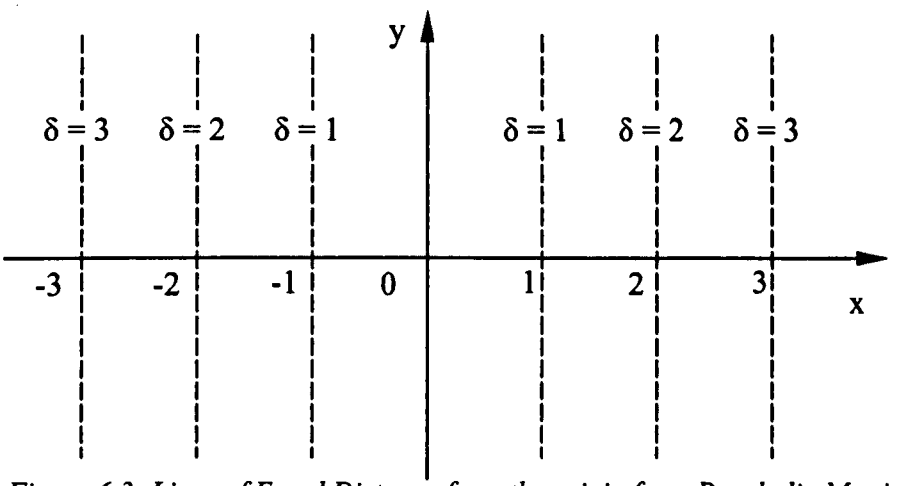


Figure 6.3: Lines of Equal Distance from the origin for a Parabolic Metric

6.1.2.3 Hyperbolic (Pseudo-Euclidean) Metric in 2D Space

In the case of the Hyperbolic (Pseudo-Euclidean) metric, a typical example of a defining expression is obtained from Equation 6.4 by setting the sign of the leading term to +1, and the other signs to –1, yielding Equation 6.7:

$$\delta = \{(x_2 - x_1)^2 - (y_2 - y_1)^2\}^{1/2} \quad \text{..... (6.7)}$$

Evaluation of this expression provides the plot of lines of equal distance from the origin shown in Figure 6.4.

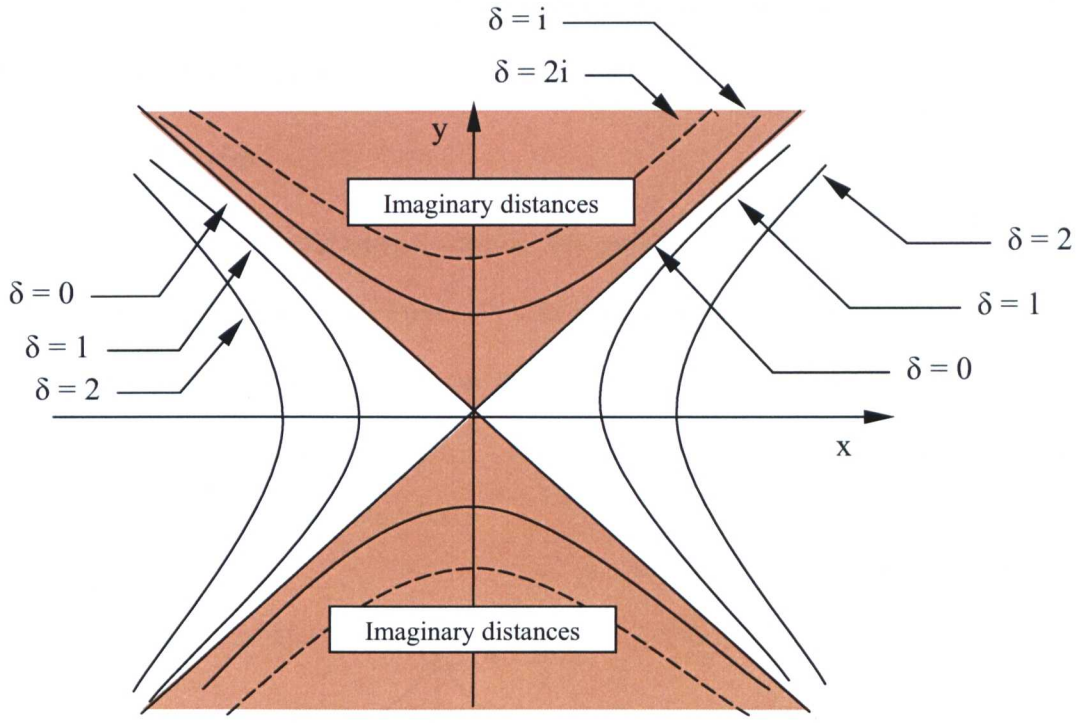


Figure 6.4: Lines of Equal Distance from the Origin for a Hyperbolic (Pseudo-Euclidean) Metric

Examination of Figure 6.4 shows that in the case of this Hyperbolic (Pseudo-Euclidean) metric, the lines (hyperbolae) of equal distance from the origin occur in 'pairs', there being one branch of the line on each side of the origin, with these being bounded by the $\delta = 0$ lines which lie at 45° to the axes. Evaluation of simple cases also shows that those lines of equal distance lying in the upper and lower zones bounded by the $\delta = 0$ lines, ie when $(x_2 - x_1)^2 < (y_2 - y_1)^2$, have complex values for their distances from the origin.

6.2 Uniqueness of Distances Between Points

It was shown in Chapter 3, that systems can be represented in ways that lend themselves directly to mathematical manipulation. The way in which these representations can be used as inputs to the calculation of ISDs will be discussed later in this section. Firstly, it is appropriate to consider in greater detail the nature of the ISD metrics that have been introduced, and in particular, the uniqueness of the distances calculated. The reason why this is especially important is that the notion of ISD is predicated upon the intuitive assumption that individual systems are distinct from one another, that two distinct points represent two different systems, and that there is a non-zero distance between them. It is shown in the following sub-sections that this is not to be taken for granted. The question may be asked:

'Is it possible for two points to be the same distance from the origin, be zero distance from each other, and yet not be coincident?'

6.2.1 Linear Metrics

In the case of a linear (Taxicab) metric, consider two points, (1) and (2) lying on the $\delta = 4$ line of equal distance from the origin (see Figure 6.1). In Figure 6.5, treat point (1) as a new origin, O′, and construct a set of lines of equal distance from O′. Consideration shows that point (2) can only be the same distance from the first origin, O, as point (1), and zero distance from the new origin, O′, if point (1) and point (2) are coincident:

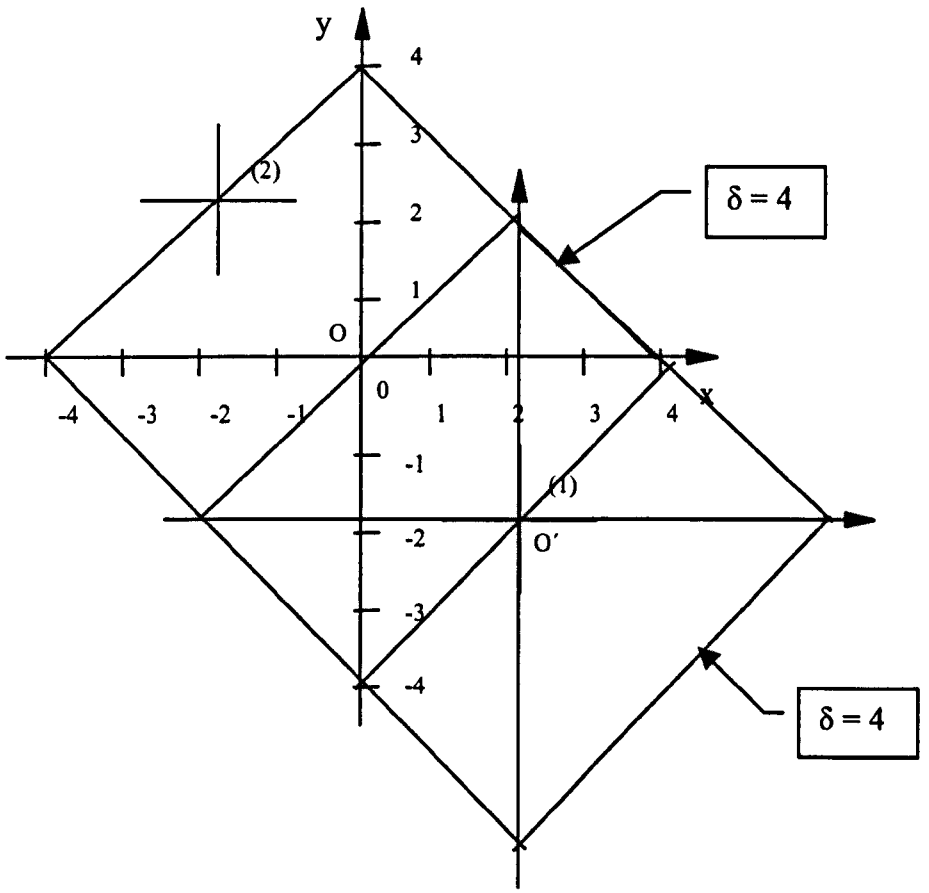


Figure 6.5: Uniqueness of Points in a Linear (Taxicab) Metric

Therefore:

> **In a Linear (Taxicab) metric,** any two points at the same distance from the origin, and zero distance from each other, have to be coincident.

6.2.2 Quadratic Metrics

In the case of quadratic metrics, the situation is more complicated. Each of the three types discussed previously is examined in more detail.

6.2.2.1 Elliptic (Euclidean) Metric in 2D Space

Consider the lines of equal distance from the origin, O, illustrated in Figure 6.2, and two points, (1) and (2), lying on the $\delta = 3$ circle, ie at the same distance from O (see Figure 6.6). Treat one of these points (1) as a new origin, O′, and construct a second set of distance circles (δ'), centred on O′ (shown dotted):

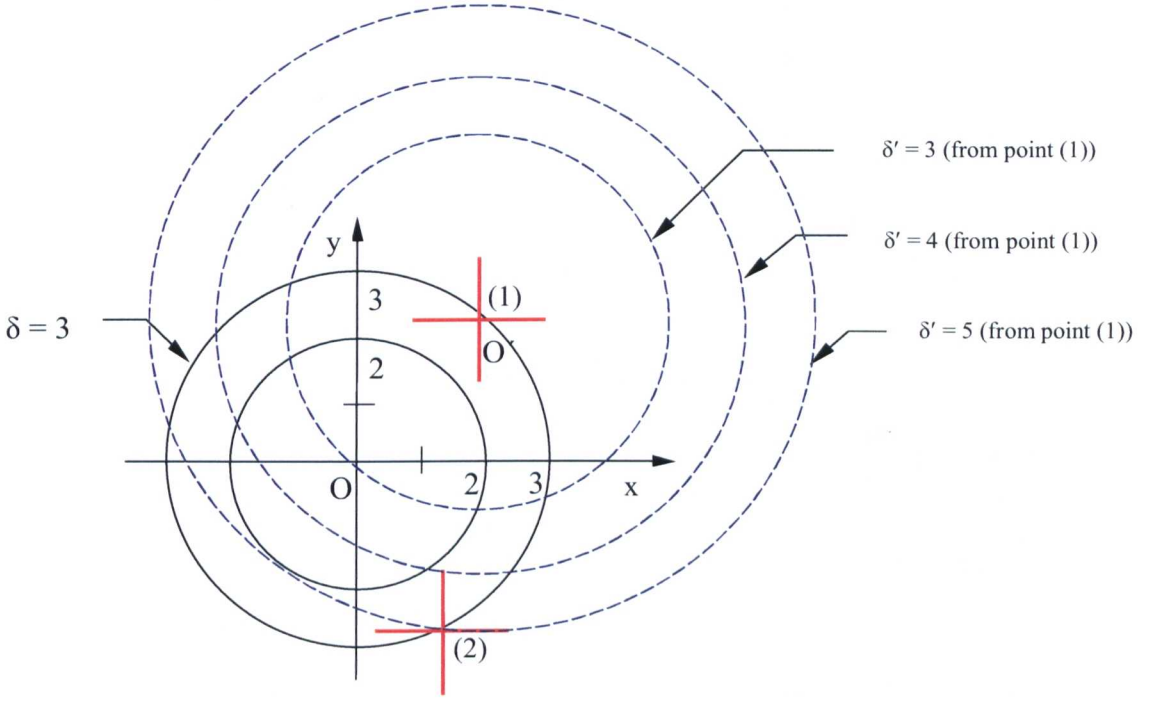


Figure 6.6: Uniqueness of Points in an Elliptic (Euclidean) Metric

Inspection of Figure 6.6 shows that point (2) lies at distance $\delta' = 5$ from point (1), and that if it were at distance $\delta' = 0$ from point (1), then it would have to be coincident with point (1). Therefore:

> **In an Elliptic (Euclidean) metric**, any two points at the same distance from the origin, and zero distance from each other, have to be coincident.

6.2.2.2 *Parabolic Metric in 2D Space*

Consider the lines of equal distance from the origin, O, illustrated in Figure 6.3 for a Parabolic metric, and two points, (1) and (2), lying on one of the $\delta = 2$ lines, ie at the same distance, 2, from O (see Figure 6.7). It can be seen that these two points are zero distance from each other, because they both have the same x coordinate. Construct a new set of axes centred at point (1), with origin O′. In this case, it can be seen that point (2) is zero distance ($\delta' = 0$) from O′, because both point (1) and point (2) have the same x coordinate ($x = 0$ in the new axis system).

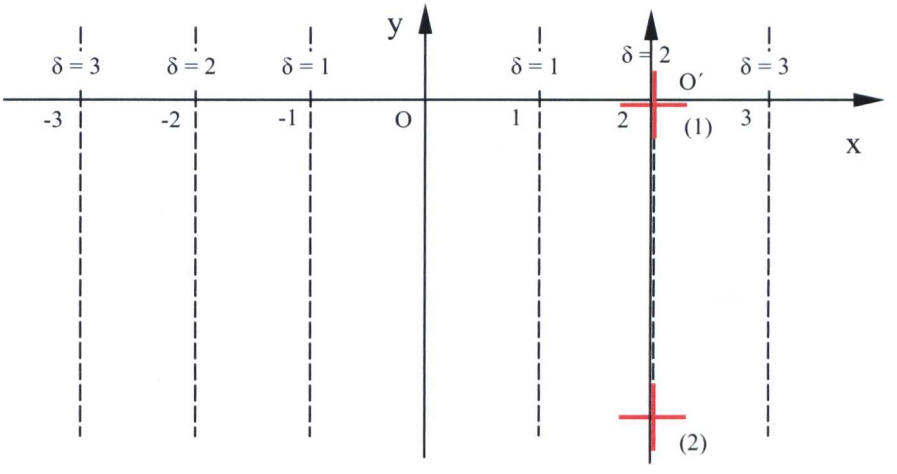


Figure 6.7: Non-Uniqueness of Points in a Parabolic Metric

Therefore:

> **In a Parabolic metric**, it is possible for two distinct points to be the same distance from the origin, and to be either zero distance from each other, or twice the distance from the origin from each other.

6.2.2.3 *Hyperbolic (Pseudo-Euclidean) Metric in 2D Space*

Consider the lines of equal distance from the origin, O, illustrated in Figure 6.4, and two points, (1) and (2), lying on one of the $\delta = 2$ hyperbolae, ie at the same distance, 2, from O (see Figure 6.8). Treat one of these points as a new origin O′, and construct a second set of distance hyperbolae (δ') centred on O′ (shown dotted).

Considering point (2), this is the same distance ($\delta = 2$) from the origin, O, as point (1). For any points to be at zero distance from (1), they must lie on the $\delta' = 0$ lines for point (1) (bold dashed in Figure 6.8). Since there are no intersections between the point (1) zero distance from O′ lines and

the $\delta = 2$ lines from the origin, O (bold chain-dotted in Figure 6.8), apart from the point (1) origin, O, it follows that the only way that point (2) can be zero distance from point (1) is for it to be coincident with point (1). Note, however, that if point (2) and point (1) both lie on the $\delta = 0$ from O (45°) lines, they will not necessarily be coincident, and will also lie on the $\delta' = 0$ from O′ (45°) lines.

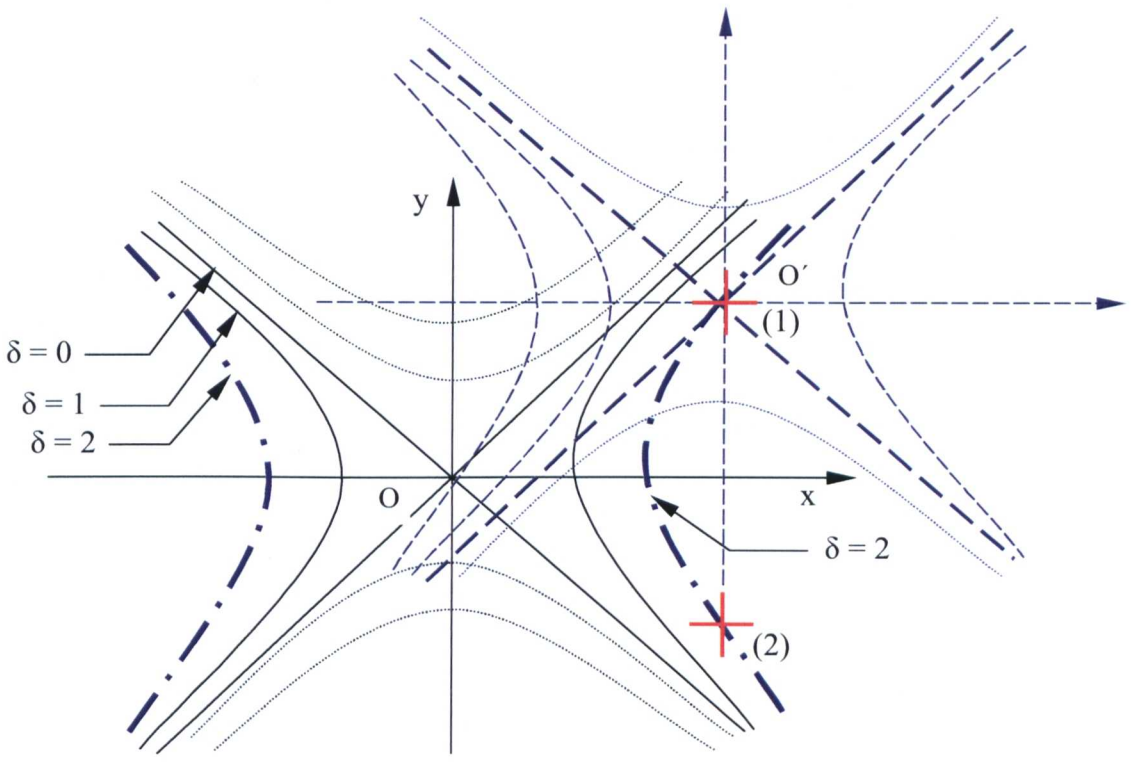


Figure 6.8: Uniqueness of Points in a Hyperbolic (Pseudo-Euclidean) Metric

Therefore:

> **In a hyperbolic metric**, any two points at the same distance from the origin, and zero distance from each other, have to be coincident, or else lie on 45° lines, ie have equal x and y coordinates.

6.3 Evaluation of Metrics

Based on the foregoing definitions, metrics were evaluated for each of the four types (taxicab, elliptic, parabolic and hyperbolic) discussed, for the 'typical set' of systems defined at the start of Chapter 5. The input data used in this evaluation is detailed in Tables F1 to F3 in Appendix F. These tables identify the metrics examined, the input data and formulae applied, and the

assumptions made. The results on which later comments are based are presented in Tables F4-F11, of the same Appendix.

6.4 Directionality of Inter-System Distance

So far, the discussion of ISD has concentrated on the distance separating two systems (points). In addition to this distance, it is possible to define a corresponding directional relationship between the two points – for example, in 2D space, the relationship between two systems can be viewed in navigational terms as requiring the definition of both range and bearing (an angle).

Two systems are represented by the points (x_1, y_1) and (x_2, y_2), with position vectors $\underline{r}_1$ and $\underline{r}_2$ respectively, relative to an origin, O (Figure 6.9). The angle, θ, between the two position vectors, can be defined in terms of the scalar (or dot) product of the two vectors. In terms of the coordinates of the points, the dot product is defined using the metric (here elliptic) as:

$$\underline{r}_1.\underline{r}_2 = x_1x_2 + y_1y_2$$

with $$\underline{r}_1.\underline{r}_1 = x_1^2 + y_1^2$$

and $$\underline{r}_2.\underline{r}_2 = x_2^2 + y_2^2$$

The angle, θ, between $\underline{r}_1$ and $\underline{r}_2$ is given by:

$$\underline{r}_1.\underline{r}_2 = |\underline{r}_1|.|\underline{r}_2| \cos\theta$$

where $|\underline{r}_1| = \sqrt{(\underline{r}_1.\underline{r}_1)}$, and $|\underline{r}_2| = \sqrt{(\underline{r}_2.\underline{r}_2)}$

Then, if $0 < \theta < \pi$: $$\theta = \cos^{-1} \{(x_1x_2 + y_1y_2) / \{\sqrt{(x_1^2 + y_1^2)}.\sqrt{(x_2^2 + y_2^2)}\} \quad \ldots\ldots\ldots (6.8)$$

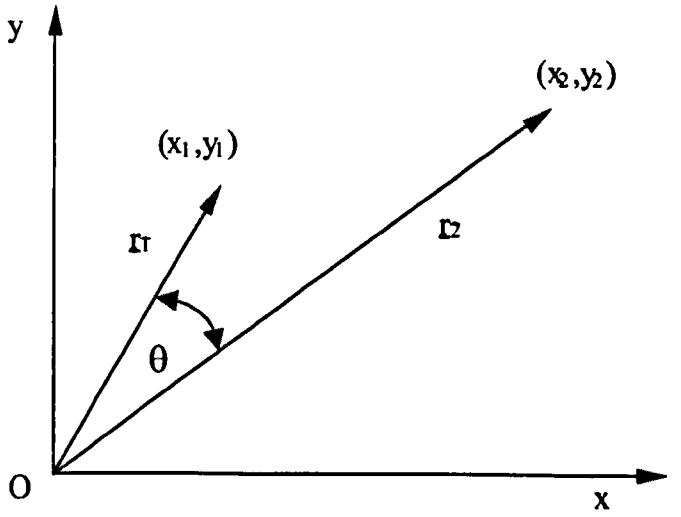


Figure 6.9: Relative Directions of Two Position Vectors in 2D Space

Equation (6.8) can be extrapolated to 3 dimensions, giving:

$$\underline{r}_1.\underline{r}_2 = x_1x_2 + y_1y_2 + z_1z_2$$

with $\underline{r}_1.\underline{r}_1 = x_1^2 + y_1^2 + z_1^2$

and $\underline{r}_2.\underline{r}_2 = x_2^2 + y_2^2 + z_2^2$

then:

$$\theta = \cos^{-1} \{(x_1x_2 + y_1y_2 + z_1z_2) / \sqrt{(x_1^2 + y_1^2 + z_1^2).\sqrt{(x_2^2 + y_2^2 + z_2^2)}}\} \ldots\ldots\ldots (6.9)$$

The expression Equation (6.9) is valid for extrapolation to n dimensions. As an example, the angle, θ, between the position vectors is evaluated in 4 dimensions for the systems in Appendix C, Appendix D, and Appendix E, using an Elliptic (Euclidean) metric with system 2.0.2 as the first (or reference) vector, $\underline{r}_1$.

Here, the space is 'eigenvalue space' where the four components of each position vector are the four eigenvalues of the characteristic polynomial derived from the adjacency (or constraints) matrix of the system being represented by the position vector (see Chapter 5).

The results are presented in Table 6.1.

System	λ1	λ2	λ3	λ4	ISD Relative to system 2.0.2	Angle, θ, Relative to system 2.0.2 (degrees)	Angle in xy plane relative to system 2.0.2 (degrees)	Angle in xz plane relative to system 2.0.2 (degrees)	Angle in xw plane relative to system 2.0.2 (degrees)
2.0.2	6	6							
2.1.2	1	11			7.07	39.8	39.8	0.0	0.0
2.1.3	0	12			8.49	45.0	45.0	undefined	undefined
2.1.4	2	10			5.66	33.7	33.7	0.0	0.0
2.1.5	1	11			7.07	39.8	39.8	0.0	0.0
2.1.6	0	12			8.49	45.0	45.0	undefined	undefined
2.1.7	3	9			4.24	26.6	26.6	0.0	0.0
2.1.8	2	10			5.66	33.7	33.7	0.0	0.0
2.1.9	1	11			7.07	39.8	39.8	0.0	0.0
2.1.10	0	12			8.49	45.0	45.0	undefined	undefined
3.1.2	13.071	6	-1.071		7.15	20.8	20.3	4.7	0.0
3.1.3	13.81	6	-1.81		8.02	22.5	21.5	7.5	0.0
3.1.4	11.657	6	0.343		5.67	17.8	17.8	1.7	0.0
3.1.5	12.403	6	-0.403		6.42	19.3	19.2	1.9	0.0
3.1.6	6	13.211	-1.211		7.31	21.1	20.6	11.4	0.0
3.1.7	10.243	6	1.757		4.59	16.8	14.6	9.7	0.0
3.1.8	6	11	1		5.10	17.0	16.4	9.5	0.0
3.1.9	6	11.831	0.169		5.83	18.1	18.1	1.6	0.0
3.1.10	6	12.708	-0.708		6.75	19.9	19.7	6.7	0.0
3.2.2	1	1	16		17.49	84.9	0.0	86.4	0.0
3.2.3	1.319	-5.35E-15	16.681		18.33	86.8	45.0	85.5	0.0
3.2.4	2	2	14		15.10	78.6	0.0	81.9	0.0
3.2.5	2.315	1	14.685		15.94	80.9	21.6	81.0	0.0
3.2.6	2.597	1.645E-15	15.403		16.88	83.2	45.0	80.4	0.0
3.2.7	3	3	12		12.73	70.5	0.0	76.0	0.0
3.2.8	3.31	2	12.69		13.57	73.6	13.9	75.4	0.0
3.2.9	3.576	1	13.424		14.53	76.6	29.4	75.1	0.0
3.2.10	3.804	1.73E-15	14.196		15.57	79.5	45.0	75.0	0.0
4.1.2	9.09	14.09	2.91	-2.09	9.37	17.1	12.2	17.8	12.9
4.1.3	8.831	14.831	3.169	-2.831	10.20	19.7	14.2	19.7	17.8
4.1.4	8.472	12.472	3.528	-0.472	7.79	17.1	10.8	22.6	3.2
4.1.5	8.217	13.217	3.783	-1.217	8.53	19.3	13.1	24.7	8.4
4.1.6	8	4	14	-2	14.42	59.5	18.4	60.3	14.0
4.1.7	7.854	10.854	4.146	1.146	6.75	19.9	9.1	27.8	8.3
4.1.8	7.606	11.606	4.394	0.394	7.31	21.1	11.8	30.0	3.0
4.1.9	7.405	4.595	12.405	-0.405	12.57	56.0	13.2	59.2	3.1
4.1.10	7.243	4.757	13.243	-1.243	13.42	57.7	11.7	61.3	9.7
4.2.2	6	16	6	-4	12.33	33.0	24.4	45.0	33.7
4.2.3	6.475	5.525	16.525	-4.525	17.15	63.7	4.5	68.6	34.9
4.2.4	6	14	6	-2	10.20	31.0	21.8	45.0	18.4
4.2.5	5.531	6.469	-2.531	14.531	14.76	60.1	4.5	24.6	69.2
4.2.6	6.877	5.123	15.123	-3.123	15.49	61.3	8.3	65.5	24.4
4.2.7	6	6	12	0	12.00	54.7	0.0	63.4	0.0
4.2.8	6.459	5.541	12.541	-0.541	12.57	56.0	4.4	62.8	4.8
4.2.9	6.838	5.162	13.162	-1.162	13.27	57.4	8.0	62.5	9.6
4.2.10	7.146	4.854	13.854	-1.854	14.07	58.9	10.8	62.7	14.5

Table 6.1: ISD Values and Directions for the Eigenvalues Elliptic (Euclidean) Metric

Note: if one or both position vectors have zero length, defined by the metric, ie are zero vectors, then θ is undefined. In the case of an Elliptic (Euclidean) vector, this requires all the eigenvalues of one or both vectors to be zero.

6.5 Suitability of Different Metrics for Comparing Systems

Earlier chapters have shown how several methods of representing systems can be envisaged. Two primary candidates discussed are:

- Characteristic polynomial coefficients – where a system is represented as a point in n-dimensional space, whose coordinates are the ratios of the coefficients of the characteristic polynomial of the Adjacency (or Constraints) Matrix. For all but the simplest systems, there will be (n+1) coefficients, and hence the identification of those that are significant is important.

- Characteristic polynomial eigenvalues – where a system is represented as a point in n-dimensional space, whose coordinates are the eigenvalues of the characteristic polynomial of the Adjacency (or Constraints) Matrix. Thus, the number of dimensions in which the system is considered to be embedded is equal to the number of links in the system under consideration.

Other representations can be envisaged, based on other permutations of the kinematic geometry characteristics discussed in Section 3.2, and these will be discussed and developed in Chapter 10 and Chapter 11.

It has now been shown that these types of representation can be used to derive Inter-System Distances (ISDs) and corresponding directional information. Whether the choice of metric is likely to be significant, and whether this significance is likely to vary according to the nature of the representation being employed, is discussed in the following sections. The discussion is based on the results presented in Appendix F.

6.5.1 Contribution of Metrics to the Comparison of Systems

Taking the points discussed so far in this chapter together with Chapter 3 and Chapter 5 shows that potential exists to define a strategy of system representation and distance measures that will provide a useful tool for the comparison of systems. The validity of this view is now examined in greater depth.

(Note that the remainder of this chapter uses ISD values for the set of systems characterised in Appendix C, Appendix D, and Appendix E – 'the data set'. The calculation of these ISD values is detailed in Appendix F).

Chapter 5 discussed ways in which progressive system degradations due to faults can be visualised as a 'graph of system graphs', in other words as points in a multi-dimensional space, and as

groupings of systems lying on a 'fault path' and related to one another by progressive fault action. With the facility of the Inter-System Distance, additional, enhanced visualisations of the data set are achieved.

Figure 6.10, which shows a 3D projection of an n-D space, illustrates how the data set can be visualised as a series of 'nebulae' of points in space, located by the eigenvalues which are used as the point coordinates. The figure also shows how the concepts of distance and direction in ISD provide a means of understanding the relationship of one system to another. In the figure, distance and directions are given relative to an origin, but these can be given relative to any defined point. A convenient means of quoting ISDs for particular systems is by reference to some chosen system that has been designated as the datum point, ie the chosen origin.

For simplicity of presentation, Figure 6.10 presents only a three-dimensional view / projection of the members of the data set. As commented previously, it is emphasised that this is only a representation in 3-dimensional space, using a limited data set, of what is, in reality, an n-dimensional problem.

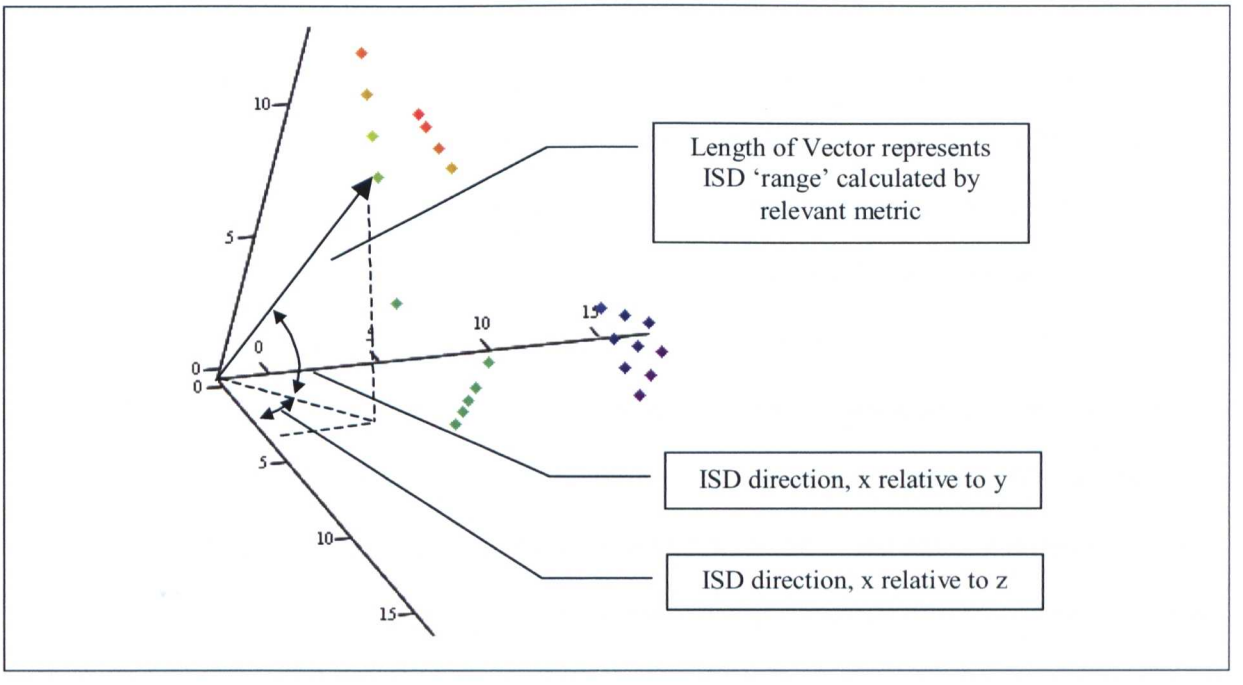


Figure 6.10 – Nebulae of 2 and 3-Bar Systems showing ISD range and direction concepts

Further insight into the detail of Figure 6.10 can be obtained by examination of the various data subsets in isolation, and this is carried out in the following section which considers how the data in Figure 6.10 can be reviewed using the various ISD concepts described. (Refer to Table 6.1 for the source data used in creating Figure 6.10).

6.5.2 Evaluation of Some Typical Metrics

6.5.2.1 2D Systems

Table 6.1 includes the ten 2D systems from the database under consideration. Reference to Appendix C shows that, when the principles of fault path definition developed in Chapter 5 are applied to these ten systems, they can be grouped into fault paths as follows:

- Fault path 1: System 2.1.2 → System 2.1.3
- Fault Path 2: System 2.1.4 → System 2.1.5 → System 2.1.6
- Fault Path 3: System 2.1.7 → System 2.1.8 → System 2.1.9 → System 2.1.10

Systems with the same colour coding are cospectral. (For further consideration of the nature of these systems, including their isomorphism characteristics, see 9.4).

(System 2.0.2, which is a two-bar, disconnected system is used as the reference point for the 2D discussion here, unless stated otherwise).

When the ISDs for these ten 2D systems are evaluated using the metrics defined earlier in this chapter, the values presented in Table 6.2 are obtained. The values are derived by evaluating the expressions for the different metrics discussed, based on two approaches. The first approach uses the system eigenvalues as position vector components (left-hand side of the table), and the second uses Characteristic Polynomial coefficients as the basis for the evaluation (right-hand side of the table). The system eigenvalues are λ_1 and λ_2 , and the characteristic polynomial coefficients are c_1 and c_2. In the latter case, since the coefficient of the term with the highest power in the polynomial will always be unity, the decision is taken to disregard the $c_0 = 1$ coefficients, and to use the two remaining coefficients as the two components of position vectors. This topic is revisited shortly, when 3D systems are discussed. Note the appearance of imaginary ISD values for the characteristic polynomial-based hyperbolic ISDs.

System	λ1	λ2	Taxicab	Elliptic	Parabolic	Hyperbolic
2.0.2	6	6	0.00	0.00	0.00	0.00
2.1.2	1	11	10.00	7.07	5.00	0.00
2.1.3	0	12	12.00	8.49	6.00	0.00
2.1.4	2	10	8.00	5.66	4.00	0.00
2.1.5	1	11	10.00	7.07	5.00	0.00
2.1.6	0	12	12.00	8.49	6.00	0.00
2.1.7	3	9	6.00	4.24	3.00	0.00
2.1.8	2	10	8.00	5.66	4.00	0.00
2.1.9	1	11	10.00	7.07	5.00	0.00
2.1.10	0	12	12.00	8.49	6.00	0.00

c1	c2	Taxicab	Elliptic	Parabolic	Hyperbolic
-12	36	0.00	0.00	0.00	0.00
-12	11	25.00	25.00	0.00	25i
-12	0	36.00	36.00	0.00	36i
-12	20	16.00	16.00	0.00	16i
-12	11	25.00	25.00	0.00	25i
-12	0	36.00	36.00	0.00	36i
-12	27	9.00	9.00	0.00	9i
-12	20	16.00	16.00	0.00	16i
-12	11	25.00	25.00	0.00	25i
-12	0	36.00	36.00	0.00	36i

Table 6.2: Table of Comparative Metrics for 2-Bar Systems

Reference to the system details in Appendix C shows that the systems represented by these data points are isomorphic – they have the same vertex connectivity - ie they are the same system [88]. This isomorphism is maintained despite the onset of faults, because although the number of degrees of freedom changes , the connectivity between the vertices is never lost.

It is, however, interesting to note that there is the possibility here for two alternative views of isomorphism – **adjacency isomorphism** and **constraints isomorphism**. The comments so far have been based on an Adjacency Matrix, **A**, view of vertex connectivity – that is to say that any type of joint is represented by a single edge. In this situation, it is true to say that the systems are isomorphic throughout the three fault paths being discussed. However, if a view of isomorphism was taken based on the Constraints Matrix, **C**, then this would mean that only certain system groupings would be isomorphic, depending on their degrees of freedom. Although not explored further here, this remains a possible area for further investigation.

In addition to being isomorphic, these same systems are also cospectral, ie they have the same Characteristic Polynomial. (Whilst cospectral systems are not necessarily isomorphic, isomorphic systems are cospectral (3.2.7).

When the data presented in Table 6.2 is plotted according to the principles outlined earlier in the chapter, the results shown in Figures 6.11 to 6.18 are obtained. An important point to note, which applies also to the 3D systems, is that each individual point plotted can represent more than one system. It will be seen from Table 6.2 (and also in Tables 6.3 and 6.4 for the 3D systems), that the various fault paths are superimposed on one another.

Cospectral systems will have the same eigenvalues, and the same characteristic polynomial coefficients, and the reason why some of these systems are superimposed in the metric plots is because they are cospectral. Consequently, the presence of cospectral systems can be seen as the underlying reason for superposition of the fault paths.

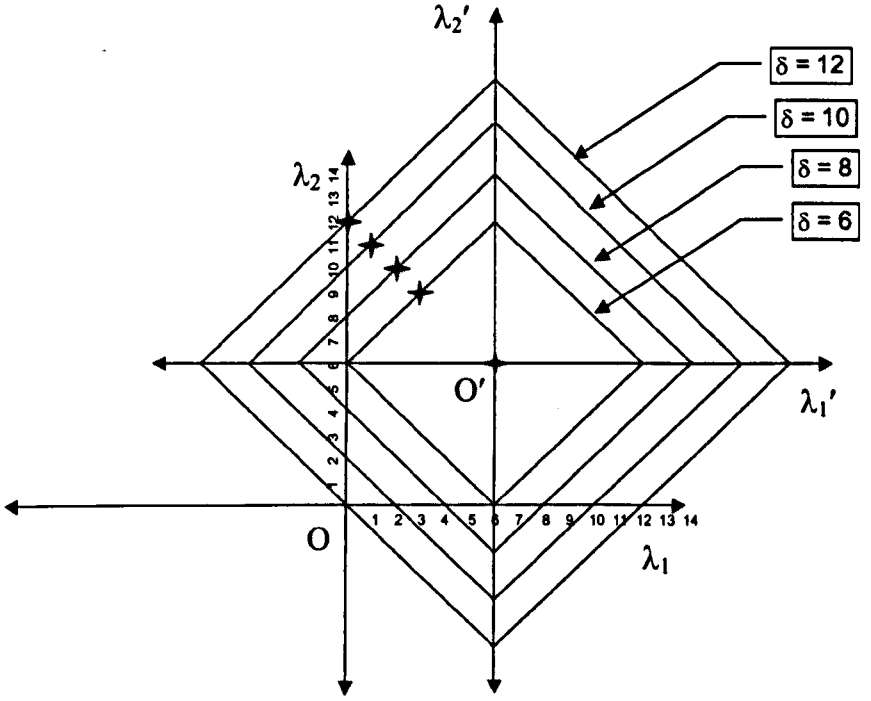


Figure 6.11 –Inter-System Distances derived for 2D Systems using a Taxicab Metric based on Position Vectors with Eigenvalue Components

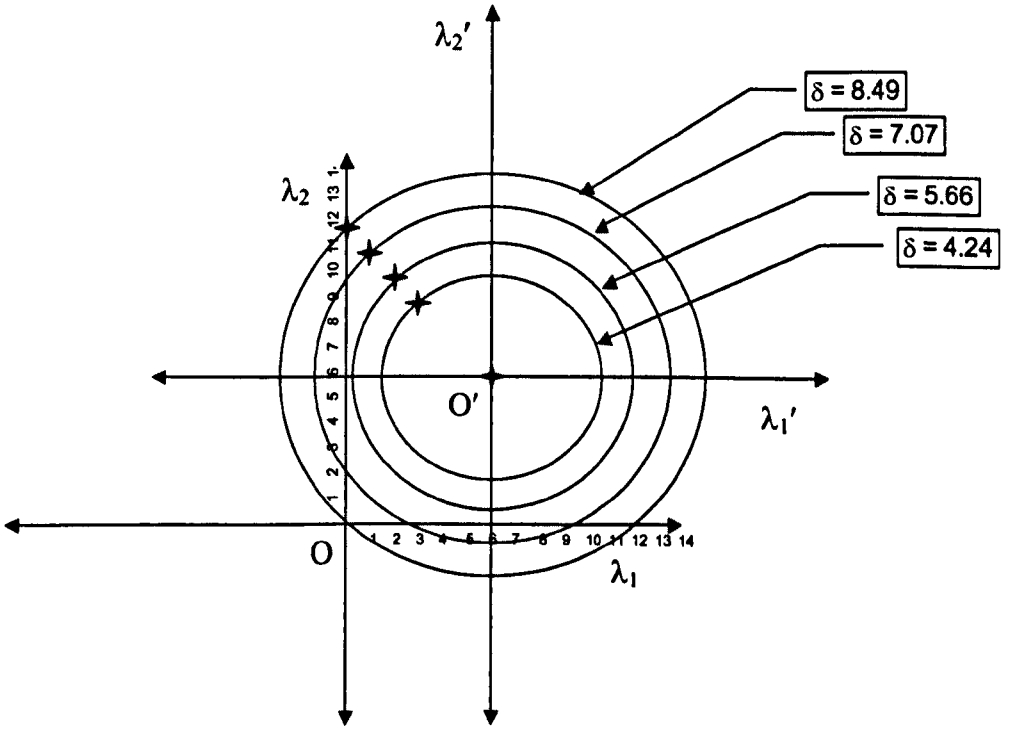


Figure 6.12 –Inter-System Distances derived for 2D Systems using an Elliptic Metric based on Position Vectors with Eigenvalue Components

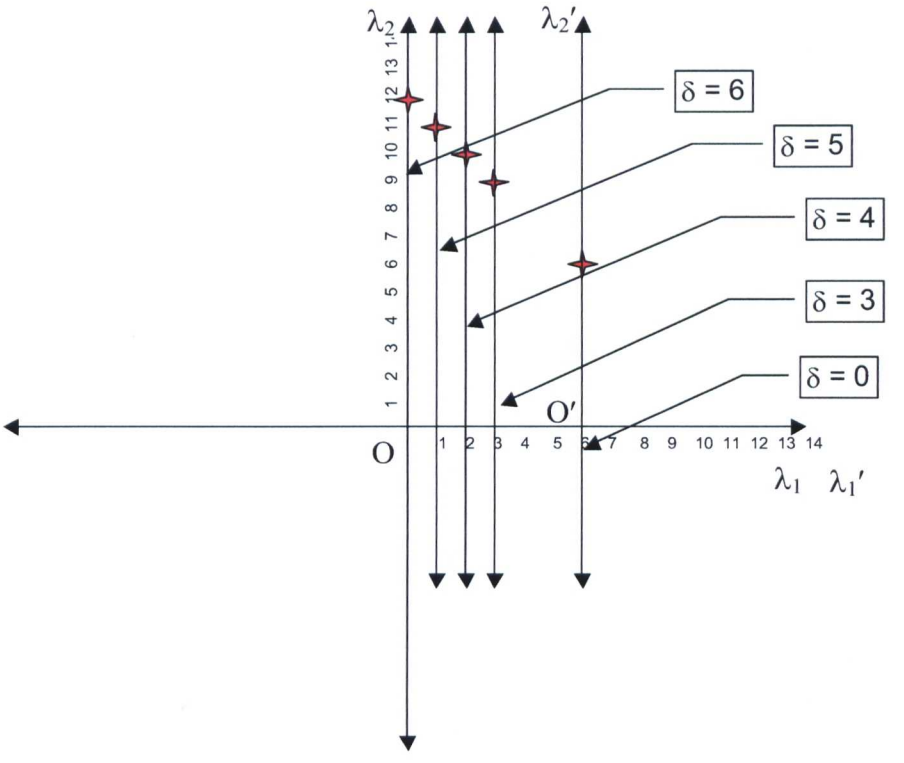


Figure 6.13 –Inter-System Distances derived for 2D Systems using a Parabolic Metric based on Position Vectors with Eigenvalue Components

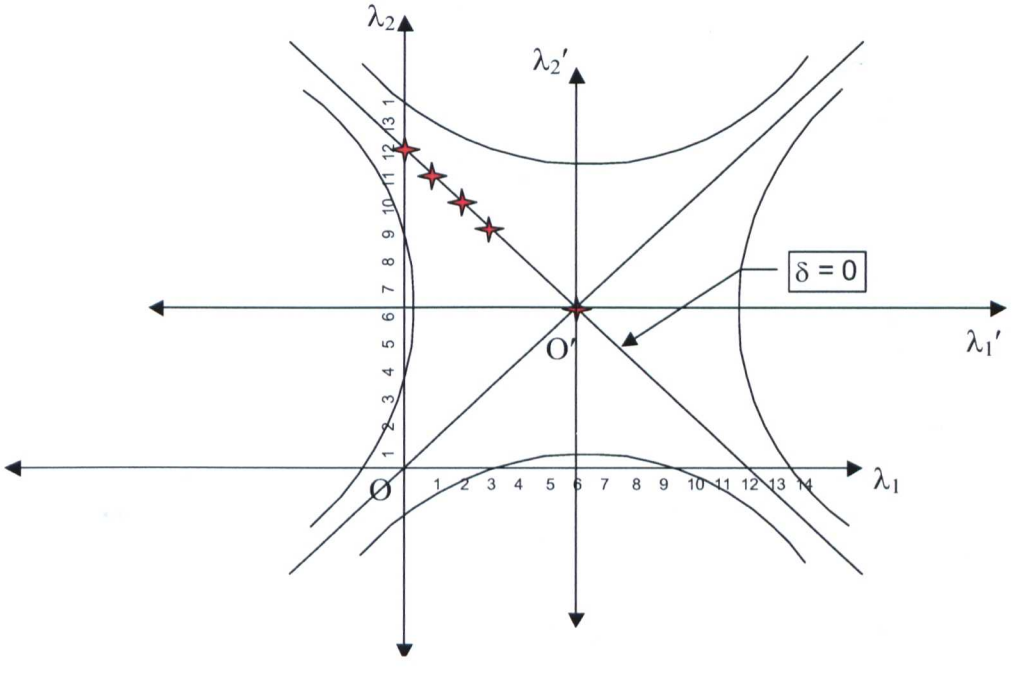


Figure 6.14 –Inter-System Distances derived for 2D Systems using a Hyperbolic Metric based on Position Vectors with Eigenvalue Components

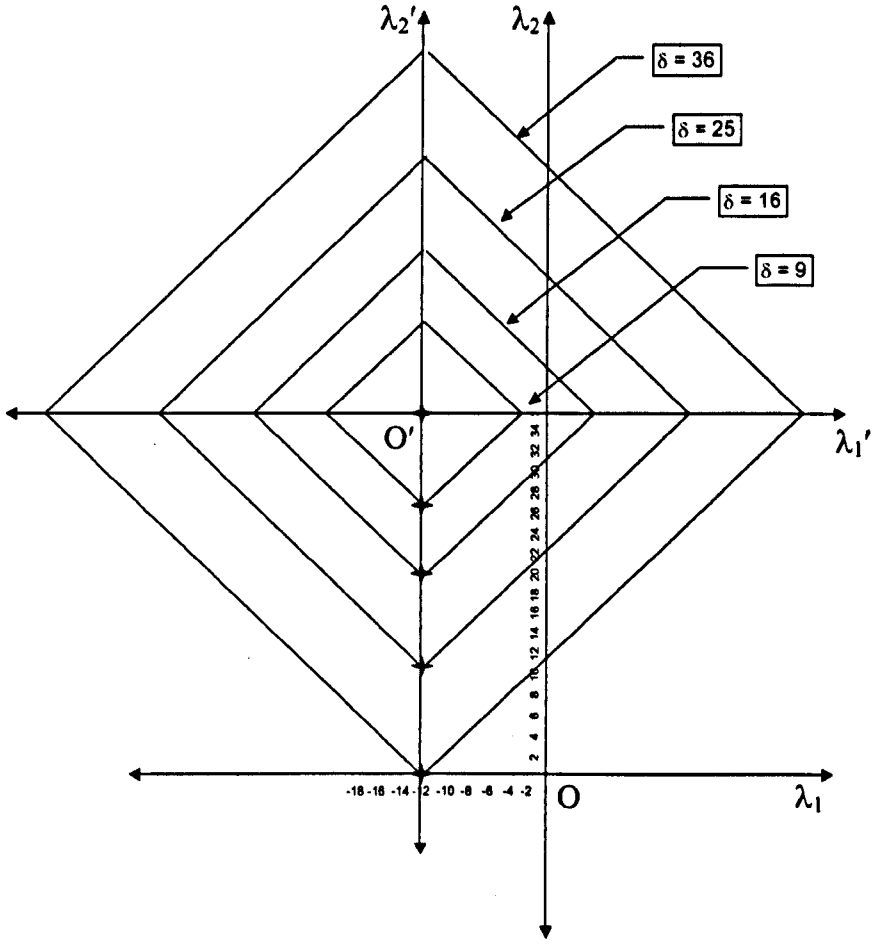


Figure 6.15 –Inter-System Distances derived for 2D Systems using a Taxicab Metric based on Position Vectors with Characteristic Polynomial Coefficient Components

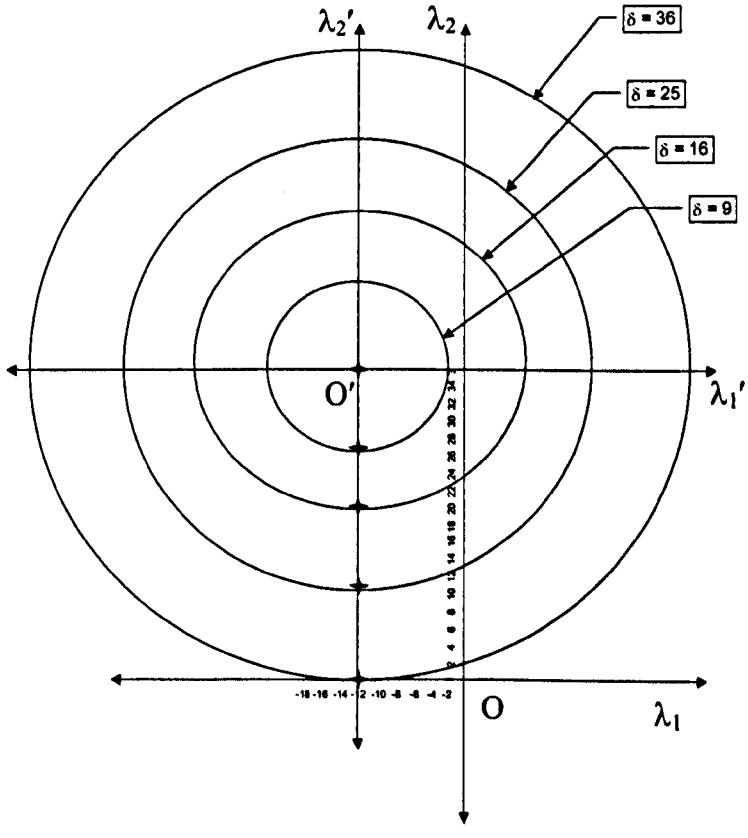


Figure 6.16 –Inter-System Distances derived for 2D Systems using an Elliptic Metric based on Position Vectors with Characteristic Polynomial Coefficient Components

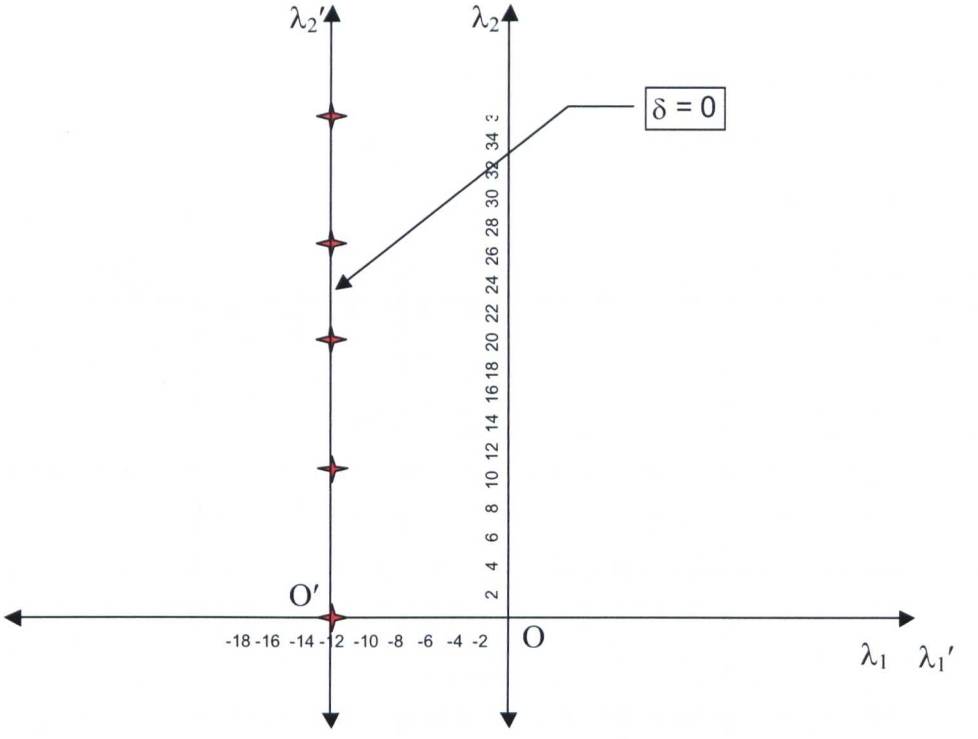


Figure 6.17 –Inter-System Distances derived for 2D Systems using a Parabolic Metric based on Position Vectors with Characteristic Polynomial Coefficient Components

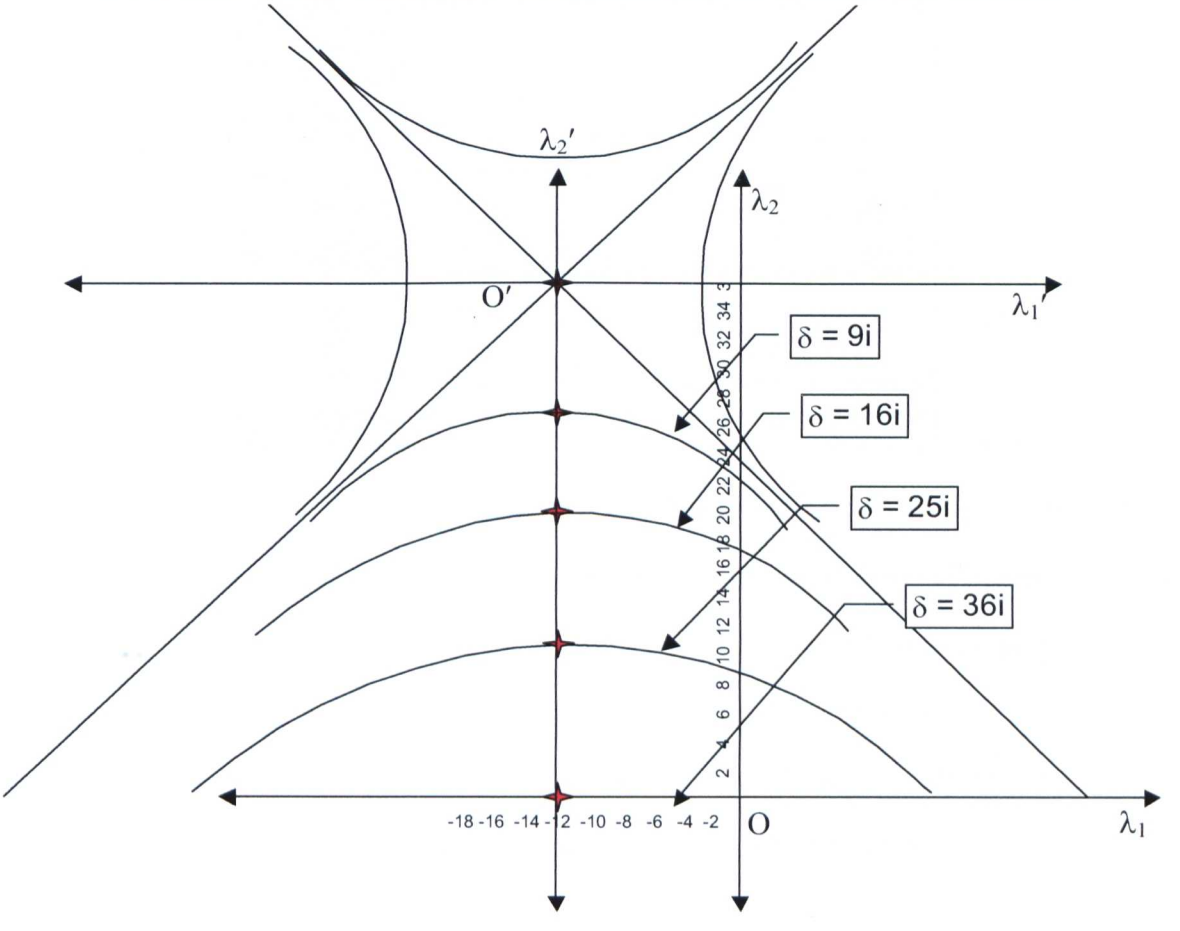


Figure 6.18 –Inter-System Distances derived for 2D Systems using a Hyperbolic Metric based on Position Vectors with Characteristic Polynomial Coefficient Components

6.5.2.2 3D Systems

The process carried out for evaluation of ISDs for 2D systems using eigenvalue and characteristic polynomial coefficient bases is repeated for the 3D systems in the database - Appendix D. The results are as presented in Tables 6.3 and 6.4.

	System	λ1	λ2	λ3	Taxicab	Elliptic	Parabolic	Hyperbolic
Fault Path 1	3.1.7	10.243	6	1.757	6.00	4.59	4.24	3.86
	3.1.8	6	11	1	6.00	5.10	0.00	5.1i
	3.1.9	6	11.831	0.169	6.00	5.83	0.00	5.83i
	3.1.10	6	12.708	-0.708	7.42	6.75	0.00	6.75i
Fault Path 2	3.1.4	11.657	6	0.343	6.00	5.67	5.66	5.65
	3.1.5	12.403	6	-0.403	6.81	6.42	6.40	6.39
	3.1.6	6	13.211	-1.211	8.42	7.31	0.00	7.31i
Fault Path 3	3.1.2	13.071	6	-1.071	8.14	7.15	7.07	6.99
	3.1.3	13.81	6	-1.81	9.62	8.02	7.81	7.60
Datum	2.0.2	6	6		0.00	0.00	0.00	0.00

Table 6.3:Four Different, Eigenvalue-Based Metrics applied to 3D Fault Path Systems

c1	c2	c3	Taxicab	Elliptic	Parabolic	Hyperbolic
-18	90	-108	162.00	120.75	0.00	$1.21i \times 10^2$
-18	83	-66	113.00	81.02	0.00	80.8i
-18	74	-12	50.00	39.85	0.00	39.4i
-18	63	54	81.00	60.37	0.00	60.07i
-18	76	-24	64.00	46.65	0.00	46.26i
-18	67	30	61.00	43.14	0.00	42.72i
-18	56	96	116.00	98.06	0.00	97.88i
-18	58	84	106.00	86.83	0.00	86.63i
-18	47	150	161.00	150.40	0.00	$1.5i \times 10^2$
-12	36		0.00	0.00	0.00	0.00

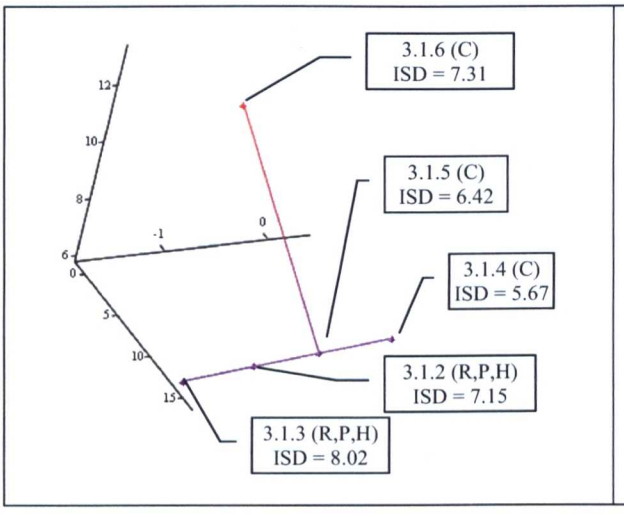
Table 6.4:Four Different, CP Coefficient-Based Metrics applied to 3D Fault Path Systems

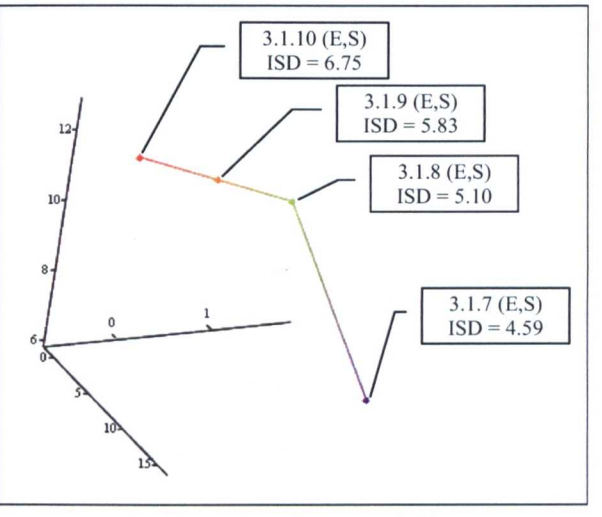
It is not attempted to plot the individual metrics for 3D systems in the same way as was done for 2D systems, since it is expected that the diagrams would be too complex to be helpful. Only two figures are provided to give some assistance with visualisation – Figures 6.19 and 6.20. These are produced using an elliptic (Euclidean) metric. Figure 6.19 shows the data for 'open' 3-Bar systems with one dof (revolute, prismatic or planar) joints and two dof (cylindric) joints.

The ISDs are again quoted relative to system 2.0.2, as for the 2D examples. The elliptic (Euclidean) metric values (and joint types) are entered in the 'callout' boxes for illustrative purposes, and reference should be made to Tables 6.3 and 6.4 for the equivalent values for the other metric types.

Figure 6.20 shows the 'nebula' for the 'open' 3-Bar systems with three dof (planar or spherical) joints, using the same notation convention as adopted for Figure 6.10. (Note that the sample data set from which the data is derived does not claim to include every permutation of these systems).

In the section covering the 2D systems, it was stated that, since the coefficient of the term with the highest power in the polynomial will always be unity, the decision was taken to suppress this coefficient and use the two remaining coefficients alone. In the 3D case, it can be shown that suppression of c_2 will have no effect on the values of the metrics. On inspection of the metric expressions, it can be seen that this will always be true in situations where the c_2 of the reference point system and that of the point being referenced to it are the same.





Note: Connecting lines are for visual reference only

Figure 6.19: 3-Bar Open Systems – types RPH & C, plotted using Eigenvalues showing an Elliptic (Euclidean) ISD

Figure 6.20: 3-Bar Open Systems – Types E,S plotted using Eigenvalues and showing an Elliptic (Euclidean) ISD

6.6 Metrics Discussion and Way Forward

The ISD values resulting from the metrics evaluations vary significantly from metric to metric, and raise some points worthy of note:

Although the points obtained are superimposed on representations of the generic metric graphs, it can be seen that this is, to an extent, an artificial construct in the cases being considered here. This is for the reasons outlined below:

Because the eigenvalues of 2D systems always lie on the straight line x + y =12, there can never be any 2D systems that lie anywhere else on the remainder of the metric graphs – ie the system points all lie on a straight line, regardless of which metric is adopted. Consequently, when compared **using a hyperbolic metric based on eigenvalues, 2D systems will always be zero distance apart.**

This also applies in the case of metrics based on characteristic polynomial coefficients. This is because c_1 will always be equal to the trace of the constraints matrix, ie 12, and so the system points will always lie on a vertical straight line. Consequently, when compared **using a parabolic metric based on characteristic polynomials, 2D systems will always be zero distance apart.**

Based on the data analysed, regardless of which metric system is adopted, **cospectral systems will always be zero distance apart from one another.** However, no cospectral systems exist in the 3D systems examined here, and this remains to be verified.

In the case of metrics generated from characteristic polynomial eigenvalues, all eigenvalues should be included in any metric evaluation, because each is equally significant. In the case of the Characteristic Polynomial coefficients, the situation is different. It is known that the leading coefficient in the Characteristic Polynomial will always be unity, and the second either zero, when derived from the Adjacency Matrix, **A**, or the value of the trace of the Constraints Matrix, **C**, if derived from **C**. It is shown that suppression of essentially fixed values does not affect ISD, provided that there is always more than one coordinate value remaining. There may be potential here to build upon the interpretation placed on the coefficients of the Characteristic Polynomial by Yan and Hall [88], and select coefficients for inclusion in the metric on the basis of their physical meaning, as defined in the reference. This is considered further in Chapter 7).

In conclusion, therefore, several methods for the description of system relationships based on metrics using position vectors formed either from the eigenvalues or the coefficients of the characteristic polynomial derived from the constraints matrix of a kinematic system's interchange graph, have been investigated. It has been shown that there is not likely to be a clear 'best metric', but that some will be more suitable than others on particular occasions, depending on what kinematic features of a system are of greatest interest at the time.

It has also been shown that comparisons based on the methods described can produce viable results for use in kinematic system representation. The fact that it is not possible to state at any one time which metric should be adopted, and that this should be assessed for any particular application and / or requirement is discussed and developed further in Section 11.2.

Chapter 7
Communication Theory, Entropy, and System Representation

Previous sections have discussed methods by which systems can be represented, and in particular, have referred to the fact that systems can be considered to lie on 'fault paths'. These faults paths are sequences of systems, where each system in the sequence is a degraded version of the previous one. This degradation is a result of the action of faults which cause fully functioning systems to evolve into systems with less functional capability, and, ultimately, to fail. By developing a way of discussing and describing such degradation, it is expected that useful insights can be found into how systems may be more faithfully represented and, ultimately, designed, so that reconfiguration in controlled ways can be used to sidestep the effects of system faults.

The impact of a fault on a system can be viewed as an increase in the state of disorder in that system in compliance with the Second Law of Thermodynamics which is stated in many ways, but including the version 'the entropy of a closed system cannot decrease'. This leads to the possibility that the methodologies developed for describing and handling entropy may be applicable to the situation under examination here.

(Note: unless otherwise stated, logarithms to base 10 are used throughout this chapter).

7.1 Entropy, Information and Probability

In his 1948 paper, Shannon [60], developed the entropy concepts established by Clausius (1822-1888)[85], Boltzmann (1844-1906)[12] and others [75] for application to communication theory. Weaver refers to a quote attributed to Boltzmann (1892), which is of particular interest and relevance here, in that Boltzmann referred to entropy as *'missing information'*, *'inasmuch as it is related to the number of alternatives which remain possible to a physical system after all the macroscopically observable information concerning it has been recorded'*. On this basis, the

relationship between physical system states, entropy and information may be established, and shows good potential for application to the representation of system fault states.

In considering this information-based view of physical systems, significant points are made by Weaver that *'information must not be confused with meaning'*, by Shannon that *'the semantic aspects of communication are irrelevant to the engineering aspects'*, and by Weaver again that *'information is a measure of one's freedom of choice (when one selects a message)'*. When this concept is applied to a physical system with a number of fault states, it can be seen that information can be considered a measure of the available fault states, any one of which can arise. As mentioned at the start of the chapter, the concept of 'fault path' is used to describe the relationship between such states (see Chapter 5 for details), and the term 'fault tree' used to describe the hierarchical representation of all available fault paths for one particular system in a single graph, or digraph – see further comments later in this chapter.

Not explored here, but worthy of note in passing are two additional aspects – firstly that this discussion leads to the interesting concept that information may be considered a 'through variable' in bond graph representations of systems, and secondly, that the evolution of system faults is a form of Markov process. Both these issues hold considerable potential for further investigation.

Shannon proved that information obeys the same laws as entropy, that is that the entropy of a set of n choices / events with probabilities of occurrence, p_1, ...p_n, may be expressed as follows:

$$H = -k \sum_{i=1}^{n} p_i \log (p_i) \quad \text{.....} \quad (7.1)$$

Where H is the entropy, p_i is the probability of an event, i, occurring, k is a scalar coefficient (in thermodynamics, this is the Boltzmann constant), and $\sum_{i=1}^{n} p_i = 1$.

In the context of the entropy of different system fault states associated via a fault path, it is possible to make the following definition:

Entropy, H, a quantity, independent of meaning and semantics, representing the uncertainty or 'lack of information' that exists regarding the selection of one particular fault path from its available fault tree – that is, it is a measure of the range of alternative configurations - 'freedom of

choice' – available to the system. It is defined quantitatively in terms of the logarithm of the number of available choices / options of system fault configurations.

The objective of this chapter is to show how some subsystem characteristics may be expressed using this entropy concept. Thus, for a single event (fault), it can be seen from Equation 7.1 that:

$$H = -k\,p \log p \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (7.2)$$

Here the number of available choices is unity – that is to say, there is no choice (of configuration / fault state or fault path), only certainty, ie p = 1.

In this condition of maximum order, it follows that:

$$H = -k\,p \log p$$

$$= -k \log 1$$

$$= 0$$

so that zero entropy is obtained for this single, certain, event to occur.

Consider now the behaviour of the relationship of Equation 7.1 when i = 2, ie when there are two possible outcomes from a situation, then with $(p_1 + p_2) = 1$ (constant k omitted):

$$H = -[\,p_1 \log p_1 + p_2 \log p_2\,],$$

and, if $p_1 = 1/3$ and $p_2 = 2/3$:

$$H = -[\,1/3 \log 1/3 + 2/3 \log 2/3\,]$$

the negative sign can be eliminated by inverting within the log expressions, yielding:

$$H = [\,1/3 \log 3 + 2/3 \log 3/2\,]$$

$$H = 0.276$$

Repeating this evaluation for six further cases corresponding to 100 % probability of one outcome or the other, equal probabilities for each outcome, and one probability being 1/4 and the other 3/4, the relationship shown in Figure 7.1 is obtained:

p_1	p_2	H
0	1	0
1/4	¾	0.244
1/3	2/3	0.276
1/2	½	0.301
2/3	1/3	0.276
3/4	¼	0.244
1	0	0

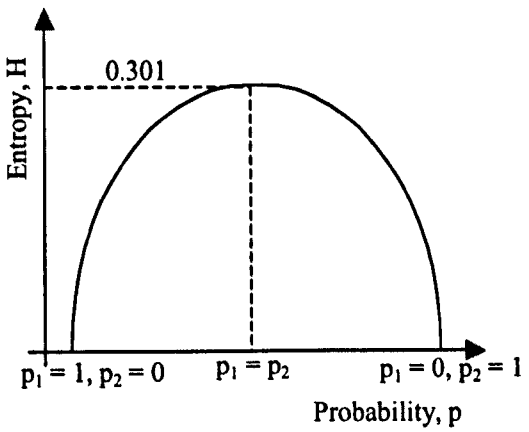


Figure 7.1: Variation in Entropy with Outcome Probabilities

Reference to the figure indicates that entropy is zero when one of the outcomes is certain and is a maximum when the outcome probabilities are equal. This is true in general. For further details of the theory and associated proof underpinning this relationship, refer to Shannon [60].

Before proceeding further, it is necessary to review three distinct entropy measures identified by Shannon. These are Total Entropy, Maximum Entropy, and Relative Entropy, defined below:

Total Entropy (H) – Shannon states *'If a choice be broken down into two successive choices, the original* H *should be the weighted sum of the individual values of* H'. Although not stated explicitly in Shannon [60], it can be shown that the weightings referred to are the probabilities of each specific outcome. In the context of the above quote, H as used here, is the Total Entropy.

Maximum Entropy (H_{max}) – If n is the number of events, then: *'For a given* n, H *is a maximum and equal to* log n w*hen all the* p_i *are equal (ie 1/n). This is also intuitively the most uncertain situation.'* [60].

Relative Entropy (H_{rel}) – *'The ratio of the entropy of a*(n information) *source to the maximum value it could have while still restricted to the same* (information) *symbols*' [60], ie H_{rel} = Total Entropy / Maximum Entropy, H / H_{max}.

7.2 Derivation of the Entropy of Simple Systems

As a simple example, consider the case of a two-bar system connected by a cylindric joint, that is to say, a joint having two degrees of freedom, and thus two potential routes for a kinematic fault to

develop. Using the notation defined in Chapter 5, the situation may be partly represented as shown in Figure 7.2, which shows one possible fault path:

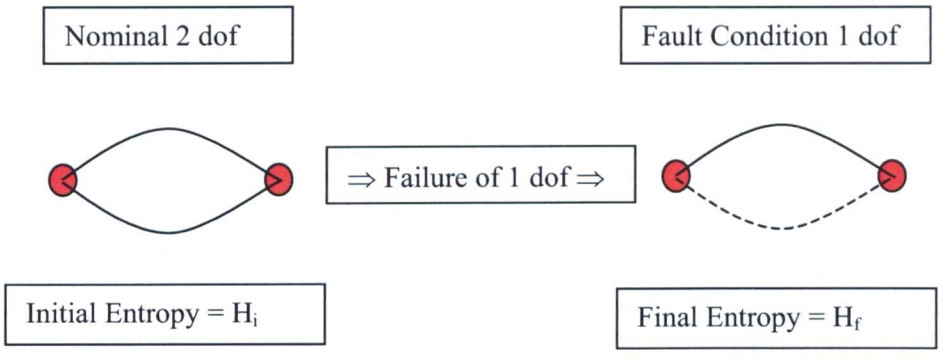
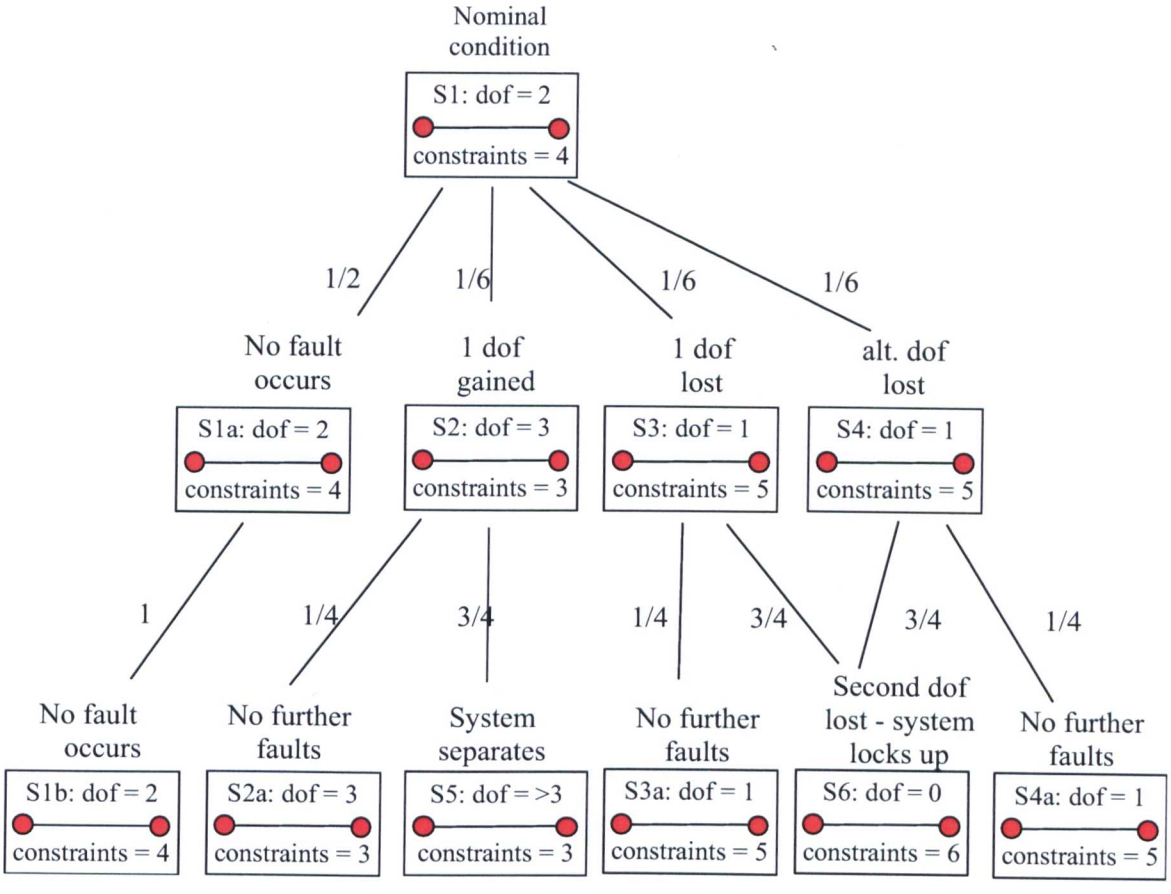


Figure 7.2:Entropy in a Two Bar System with Cylindric Joint where, say, the Rotational dof Fails

It was shown in Chapter 5 how the degradation of systems under the influence of fault action can be represented as a 'fault path' connecting the various system configurations that result from such fault action. In the majority of cases, there will be more than one possible combination of faults that can occur, and so the concept of the 'fault path' is extended to the concept of a 'fault tree'. Developing the example of Figure 7.2, based on this approach, the fault tree for the two-bar system with cylindric joint (where either dof may fail) may be represented as shown in Figure 7.3. (Only this very brief introduction to the concept of the 'fault tree', sufficient to facilitate the discussion of entropy, is given here. A more comprehensive discussion of the representation of faults is provided in Chapter 8, following).

Having established the fault tree, it is possible to apply the method used by Shannon [60 -Section 6 – 'Choice, Uncertainty and Entropy'] to calculate the entropies for the nominal state, and for the different fault conditions, based on probabilities chosen for illustration purposes. The probability values are associated with the relevant edges in the fault tree, each indicating the probability of transition along that edge.



Note: probabilities chosen are for illustration purposes only – the total of the probabilities at each level should always equal 1. In the case of the last stage, probabilities are determined as follows:

1 x 1/2	1/4 x1/6 = 1/24	3/4 x1/6 = 1/8	1/4 x1/6 = 1/24	2 x 1/6 x3/4 = 2 x 1/8	1/4 x1/6 = 1/24

$= 1/2 + 1/24 + 1/8 + 1/24 + 2(1/8) + 1/24 = 1$, as required

Figure 7.3:Fault Tree for a Two Bar System with Cylindric Joint where either the rotational dof or the translational dof may fail

7.2.1 Entropy of the Nominal (Root Node) State

Thus, referring to Node S1 (Root Node) in Figure 7.3, it can be shown that Shannon's deduction referred to earlier – '*original* H *should be the weighted sum of the individual values of* H' can be used to derive the following relationship for the original Total Entropy:

$$H = -[\ \frac{1}{2}\ \log\ \frac{1}{2}\ + \frac{1}{24}\ \log\ \frac{1}{24} + \frac{1}{8}\ \log\ \frac{1}{8} + \frac{1}{24}\ \log\ \frac{1}{24} + \frac{1}{8}\ \log\ \frac{1}{8} + \frac{1}{8}\ \log\ \frac{1}{8} + \frac{1}{24}\ \log$$

.....(7

$$\therefore H = \left[\frac{1}{2} \log 2 + \frac{3}{24} \log 24 + \frac{3}{8} \log 8 \right]$$

$$= 0.6617$$

Maximum Entropy, H_{max}, is derived from the same expression, but where each outcome is equi-probable:

$$H_{max} = -\left[\frac{1}{7} \log \frac{1}{7} + \frac{1}{7} \log \frac{1}{7} + \frac{1}{7} \log \frac{1}{7} + \frac{1}{7} \log \frac{1}{7} + \frac{1}{7} \log \frac{1}{7} + \frac{1}{7} \log \frac{1}{7} + \frac{1}{7} \log \frac{1}{7} \right]$$

.....(7.4)

Therefore:

$$H_{max} = \left[\frac{1}{7} \log 7 + \frac{1}{7} \log 7 + \frac{1}{7} \log 7 + \frac{1}{7} \log 7 + \frac{1}{7} \log 7 + \frac{1}{7} \log 7 + \frac{1}{7} \log 7 \right]$$

$$= \log 7 = 0.8451$$

..... that is to say that, in general, the Maximum Entropy is the log of the maximum number of paths at the lowest level, when equi-probable. (Note that in situations where more than one path leads to the same fault state, that fault state must be counted for each path in order to arrive at the correct value for H_{max}).

Relative Entropy = (Total Entropy) / (Maximum Entropy)

Therefore, at Root Node, Relative Entropy, $H_{rel} = 0.6617 / 0.8451 = 0.7830$

7.2.2 Entropy of the Fault States

A classical physics view of the entropy of a closed system before and after it degrades due to action of a fault might be as presented in Figure 7.4, where the initial system entropy, H, is increased during the occurrence of a fault so that the entropy of the final state, H', will comprise the entropy of the post-fault state, H_1, plus the entropy increase attributed to all the losses occurring during the fault, H_2. This latter category includes, but is not limited to, the following:

- Entropy of any separated parts which no longer form part of the final fault state *
- Entropy increases due to Friction, Noise, Heat, Stress, Strain, Elastic deformation during the fault process, and other causes *.

*Both these entropies continue to belong to the same closed system.

Thus $H' = H1 + H2$, and $H < H'$

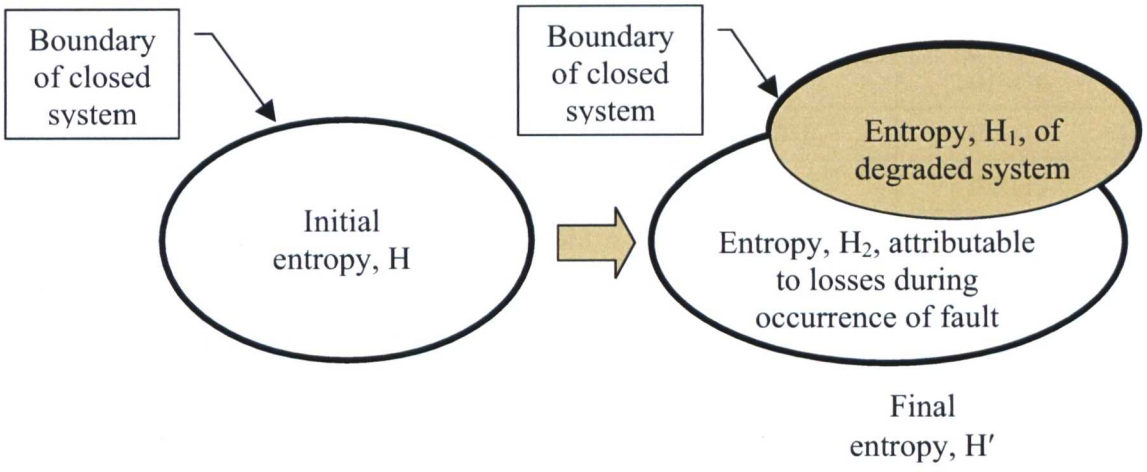


Figure 7.4: Classical Physics View of Entropy Changes with Degeneration into Fault State

However, to take an 'information based' view of such a situation requires that the entropy of the new state – the selected possibility - must be considered in the context of the other, unselected, states / possibilities. In this case, the model of Figure 7.4 needs to be modified as presented in Figure 7.5, where the initial system entropy, H, is increased during the occurrence so that the final entropy after the occurrence, H′, is numerically equal to the sum of the entropy of the selected possibility, H_1, and the entropies of the alternative, unselected possibilities, H_2.

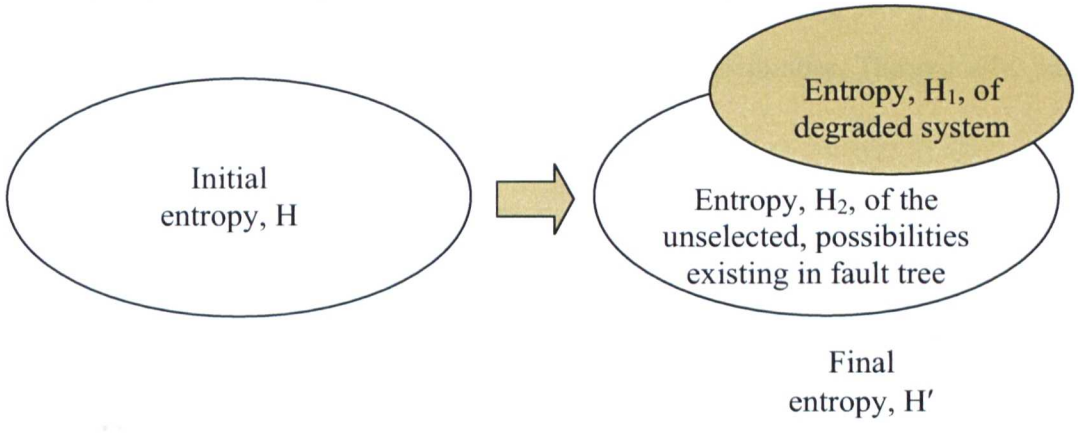


Figure 7.5: Communication Theory View of Entropy Changes with Degeneration into Fault State

The process applied earlier for determining H, H_{max}, and H_{rel} of the Root Node can now be repeated (with this viewpoint in mind) for each of the other nodes, allowing tables showing the variation of Entropy with fault evolution to be created. Two variations on such a table can be envisaged. The

first of these – Table 7.1 – is with the total, maximum and relative entropies calculated for the fault node alone, with no reference made to other potential fault states.

Node	Condition	Total Entropy, H_{tot}	Maximum Entropy, H_{max}	Relative Entropy, H_{rel}
S1	Root node	0.6617	0.8451	0.7830
S2	1 dof gained	0.2442	0.3010	0.8113
S3	1 dof lost	0.2442	0.3010	0.8113
S4	Alternative dof lost	0.2442	0.3010	0.8113
S1a	No fault occurs	0	0	indeterminate
S1b	No fault occurs	0	0	indeterminate
S2a	1 dof gained, no further faults	0	0	indeterminate
S5	System separates - complete failure	0	0	indeterminate
S3a	1 dof lost, no further faults	0	0	indeterminate
S6	System locks up	0	0	indeterminate
S4a	Alternative dof lost, no further faults	0	0	indeterminate

Table 7.1: Variation in Entropy Calculated for the Fault Node alone with Fault Evolution for a Two-Bar System with Cylindric Joint

The second option – Table 7.2 – is where the total, maximum and relative entropies of a fault node are calculated using the model outlined in the text associated with Figure 7.5. In this model it is necessary to determine the entropy, H_2, of the unselected possibilities. Theoretically, this can be done in two ways:

a) Establish the entropy of the entire fault tree with the selected option suppressed, to be H_2, or

b) Use the sum of the entropies of the unselected options to be H_2.

In (a), the problem is that once the selected option is suppressed, this will result in a complete rebalancing of the probabilities and structure of the tree, resulting in an entropy value for H_2 which does not have a relationship to the original nominal configuration. Therefore, it is considered that option (b) is the better approach, and this can be applied to the values in Table 7.1 to yield Table 7.2. This table yields an interpretation of the entropy results which is most in line with the definition of Figure 7.5, that is that the entropy increases after occurrence of a fault, falling to zero once the outcome is certain – Equation 7.2

Node	Condition	Total Entropy, H_{tot}	Maximum Entropy, H_{max}	Relative Entropy, H_{rel}
S1	Root node	0.6617	0.8451	0.7830
S2	1 dof gained	3 x 0.2442 = 0.7326	3 x 0.3010 = 0.9030	0.8113
S3	1 dof lost	3 x 0.2442 = 0.7326	3 x 0.3010 = 0.9030	0.8113
S4	Alternative dof lost	3 x 0.2442 = 0.7326	3 x 0.3010 = 0.9030	0.8113
S1a	No fault occurs	0	0	indeterminate
S1b	No fault occurs	0	0	indeterminate
S2a	1 dof gained, no further faults	0	0	indeterminate
S5	System separates - complete failure	0	0	indeterminate
S3a	1 dof lost, no further faults	0	0	indeterminate
S6	System locks up	0	0	indeterminate
S4a	Alternative dof lost, no further faults	0	0	indeterminate

Table 7.2: Variation in Entropy Calculated for the Fault Node and non-Fault Nodes Combined with Fault Evolution for a Two-Bar System with Cylindric Joint

As an additional argument in favour of the option (b) approach defined above, it will be seen that where the same fault state can be arrived at from different Root Nodes, that is to say, different systems, then interpretations of entropy which are independent of the original Root Node will be of greater use in system comparisons than other interpretations.

In conclusion, therefore, it can be stated that this chapter has shown that entropy can be used as a contribution to the representation of system behaviour. It is, however, important to note that, in most circumstances, it is not a 'stand-alone' indication. That is to say, that to understand entropy when defined in certain ways – for example in Figure 7.5, the context has to be provided at the same time. This has implications for the work later in this thesis.

Chapter 8
Classification of System Fault States

Previous chapters have shown that it is possible to represent kinematic system topologies and the kinematic faults to which they are subjected, using relationships based on the eigenvalues and eigenvectors of the characteristic polynomials, $P(\lambda)$, of the adjacency or constraints matrices of the interchange graphs of a system's nominal and fault states (Chapter 5), and also by considering the nature of entropy changes that occur as systems degrade (Chapter 7). This chapter now considers what additional information can be extracted from these characteristic polynomials, and the common features, if any, of the fault-induced degradation undergone by such kinematic systems.

8.1 Method Applied

In Chapter 3 it was shown that there was good evidence that the characteristic polynomial coefficients derived from the constraints matrix, **C** of the interchange graph of a kinematic system, showed a similar potential for establishing 'coefficients by inspection' to that established by Yan and Hall [90] for an adjacency matrix. Section 8.2 shows that this is, indeed, the case by examining the characteristic polynomials for two, three and four-bar systems obtained by symbolic expansion of the determinants of the constraints matrices for each of the systems. The results obtained when each of the symbolic terms in the expansion of the determinant is evaluated, by substitution of numerical values, are considered, and used to define the supposed 'coefficients by inspection' expression for each of the characteristic polynomial coefficients.

Section 8.3 takes five example kinematic systems, and derives for each of them a table of all relevant kinematic fault states, and then uses this table to derive a 'fault state digraph' for each of the systems. Using this data, the way in which the fault states of a system can be grouped into 'fault classes' is discussed, using representations derived specifically to draw out the similarities between

groups of fault states for a given kinematic system. Fault states are also defined using symbolically expanded characteristic polynomial determinants, allowing reference back to the first part of this chapter.

Section 8.4 introduces a symbolic convention to represent the symbolic terms in the expanded determinant, to establish a meaning for these terms, and then to extend the method to the representation of fault states.

An unfortunate complication with analysing graphs of systems under fault conditions is that it is difficult to define a universal notation that will represent all aspects of the problem. For this reason, a revised notation is introduced before proceeding further.

8.1.1 Further Development of Terminology and Graph Notation / Symbology

Any nominal or fault state of a kinematic system may be represented using an interchange graph where the edges represent the kinematic joints of the system, and the vertices represent its links. From this interchange graph, Chapter 3 showed that as well as the standard graph theoretical method of forming the adjacency matrix representing the edge interconnections between the vertices of the graph, another matrix – the 'constraints matrix' can be formed where each term of this matrix represents the number of constraints existing on the interchange graph edge represented by that term.

Reference has also been made in Chapter 3 to the definition of the Characteristic Polynomial, $P(\lambda) = |\lambda\mathbf{I} - \mathbf{C}|$ (where it is based upon the Constraints Matrix, **C**, of the kinematic system's interchange graph), or $P(\lambda) = |\lambda\mathbf{I} - \mathbf{A}|$ (where it is based upon the Adjacency Matrix, **A**, of the kinematic system's interchange graph). In either case, this is derived from the Characteristic Matrix which can be written in the following form:

$$P(\lambda) = |\lambda\mathbf{I} - \mathbf{A}|, \text{ or } |\lambda\mathbf{I} - \mathbf{C}| = \begin{vmatrix} (\lambda - d_{11}) & d_{12} & \dots & d_{1n} \\ d_{21} & (\lambda - d_{22}) & \dots & d_{2n} \\ d_{31} & d_{32} & \dots & d_{3n} \\ \dots & \dots & (\lambda - d_{ij}) & \dots \\ d_{n1} & d_{n2} & \dots & (\lambda - d_{nn}) \end{vmatrix} \quad \dots\dots\dots\dots\dots\dots (8.1)$$

$$= p_0\lambda^n + p_1\lambda^{n-1} + \dots\dots + p_n$$

Where λ is a dummy variable, d_{ij} is any element of the matrix where i (j) are row (column) indices respectively, and **I** is an n x n unit matrix.

Thus, it is possible to use the interchange graph to derive the adjacency matrix or constraints matrix of any kinematic system under consideration, and hence either to evaluate the determinants of these matrices, or the corresponding characteristic polynomials, and to use these as representations of the systems.

However, it will be understood that either of these approaches, encapsulates valuable information regarding the presence or absence of edges in the interchange graph and hence defining the detailed structure of the latter. There is, therefore, considerable value in highlighting this information in the constraints (or adjacency) matrix determinant, and this can be achieved by expanding this determinant symbolically, as shown in the simple example below:

Symbolic expansion of determinant

$$\begin{vmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \end{vmatrix} = (d_{11})(d_{22})(d_{33}) + (d_{12}d_{23}d_{31}) + (d_{13}d_{21}d_{32}) - (d_{22})(d_{13}d_{31}) - (d_{11})(d_{23}d_{32}) - (d_{33})(d_{12}d_{21})$$

where brackets are used to form terms within each of the six components of the expansion according to the schema discussed below. In order to facilitate further application of this approach, a terminology is first defined based on the form taken by the terms of the symbolic expansion of the determinant of the constraints (or adjacency) matrix of a kinematic system, described below and illustrated in Figure 8.1:

- a term in an expanded symbolic determinant derived from a typical single vertex (labelled 1) in the interchange graph (a 'self-loop), and represented symbolically as, for example, (d_{11}), is referred to as a '1-loop'.
- a term in an expanded symbolic determinant derived from a typical connected pair of vertices (labelled 1 and 2) in the interchange graph, and represented symbolically as, for example $(d_{12}d_{21})$, is referred to as a '2-loop'.
- a '1.2-loop' refers to a term in an expanded symbolic determinant composed from the product of one diagonal element (a 'self-loop') and two further elements combined as a '2-loop', as, for example $(d_{11})(d_{23}d_{32})$.
- a '2.2-loop' refers to a term in an expanded symbolic determinant composed from the product of two separated pairs of elements creating two separate '2-loops', as, for example $(d_{12}d_{21})(d_{34}d_{43})$.

- a '3-loop' refers to a term in an expanded symbolic determinant composed from the product of three elements creating a triangular loop, as, for example, ($d_{12}d_{23}d_{31}$). Note that a 3-loop can have a clockwise or anti-clockwise form.

- a '1.3-loop' refers to a term in an expanded symbolic determinant composed from the product of one diagonal element (a 'self-loop') and three further elements combined as a '3-loop', as, for example (d_{11})($d_{12}d_{23}d_{31}$).

- a '1.1.2-loop' refers to a term in an expanded symbolic determinant composed from the product of two diagonal elements ('self-loops') and two further elements combined as a '2-loop', as, for example (d_{11})(d_{44})($d_{23}d_{32}$).

- a '4-loop' refers to a term in an expanded symbolic determinant composed from the product of four elements creating a quadrilateral loop as, for example, ($d_{12}d_{23}d_{34}d_{41}$). Note that a 4-loop can have four forms - a clockwise or anti-clockwise form, plus a 'clockwise twisted' or 'anti-clockwise twisted' form.

It will be seen from the diagram (Figure 8.1), therefore, that, in the notation used, a symbolic determinant term such as ($d_{23}d_{32}$) will be shown as two separate arrows, one representing the $2 \rightarrow 3$ direction , d_{23}, and annotated 2,3, and one representing the $3 \rightarrow 2$ direction, d_{32}, annotated 3,2. The fact that the determinant term can be divided into two 'sub-terms', each indicating a direction, is the reason underlying the choice of terminology (in this case) '2-loop'. Additionally, the annotation has appended to it, in subscript parentheses, the number of constraints on the joint, (eg $3,2_{(4)}$ indicates four constraints on the joint connecting links 3 and 2).

In later parts of this chapter, and in Chapter 9, a shorthand version of this symbology is used. The simplified symbols for this are inset in the bottom right-hand corner of each cell in Figure 8.1.

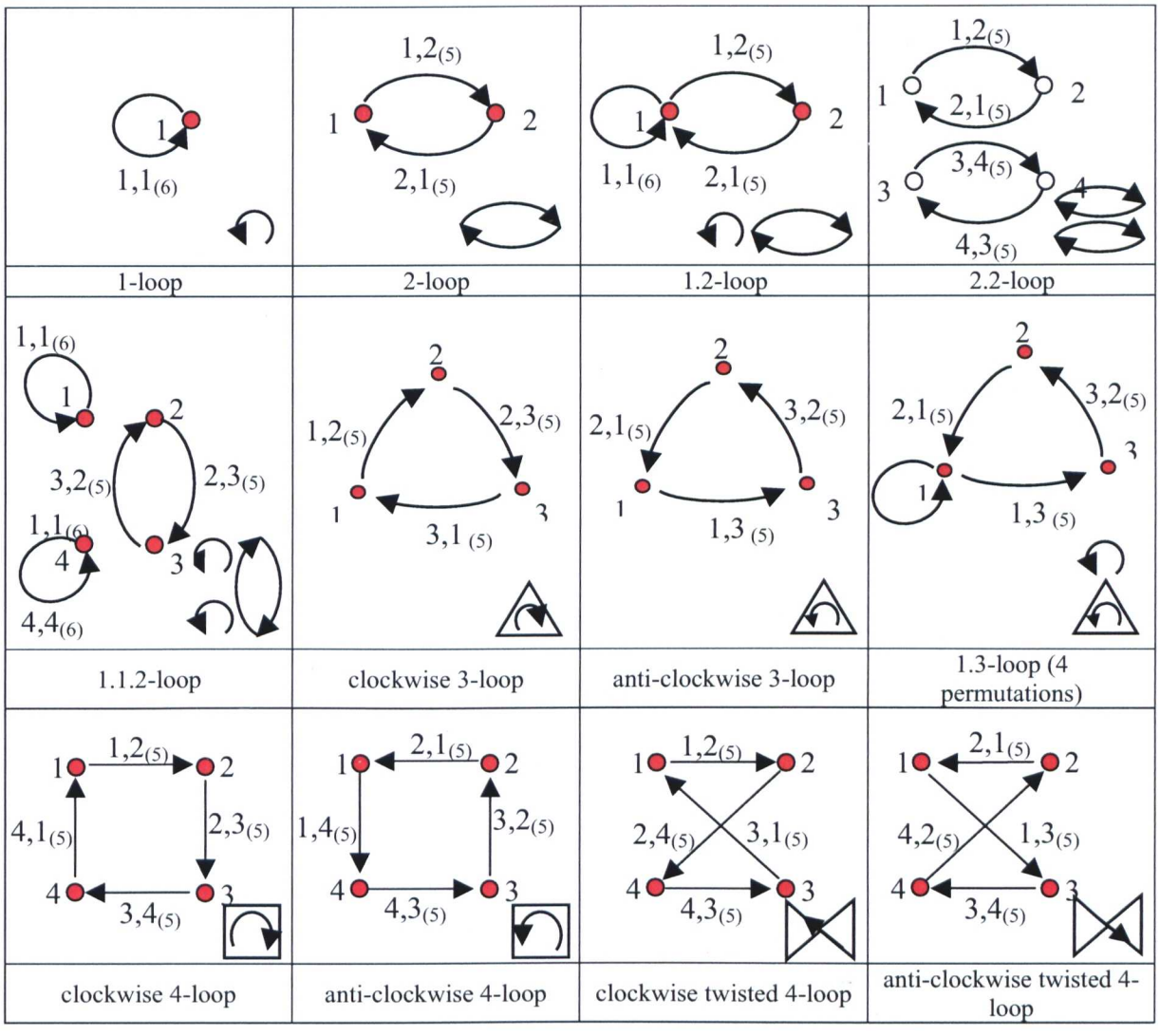


Figure 8.1: Diagrammatic Representation of a 2, 3 and 4-Loop Terminologies(with 1 dof Joints)

In Section 8.2, the large amount of data to be presented requires a more concise version of the above approach to be adopted, also, the subject matter of Section 8.2 does not require full details of the determinant terms to be stated, since it is concerned primarily with identifying available fault states. Thus, a simplified notation for indicating kinematic system joint states is adopted, based on using a simple geometric shape to represent joint type, containing a numeral indicating the number of constraints in that joint. Additionally, since two main types of graph are used here - interchange graphs and system state digraphs - different vertex symbols are adopted for each type, as shown in Table 8.1. Because this is a notation devised to meet specific presentational needs, and the examples used contain only certain 1 dof joints, not all joint types are dealt with.

Joint Type / State	Notation	
	Revolute	Prismatic
Revolute or Prismatic joint (Screw joint not used)		
Driven joint (revolute as example)		
Interchange Graph Node		
Fault State Digraph Node		

Table 8 1: Concise Joint State and Vertex Notation for use in Fault State Tables

With notation now defined, the following section proceeds to examine the nature of the determinant of some characteristic polynomials when evaluated symbolically.

8.2 Meaning of Constraints Matrix Characteristic Polynomial

8.2.1 Graph Determinants Expanded Symbolically

Since it has been demonstrated in Chapter 3 that there are significant advantages in using **C**, rather than **A**, it is considered appropriate to continue to use **C** as the basis for further investigations. In order to increase the confidence level in this assumption, it is appropriate to consider in greater detail the nature of the determinant derived from **C** , and its similarity to the accepted features of that derived from **A**. As a means of validating that solutions based on **A** and **C** are comparable, consideration is given to the 'coefficients by inspection' approach developed by Yan and Hall [90].

When $P(\lambda) = p_0\lambda^n + p_1\lambda^{n-1} + \ldots\ldots + p_n$ is expanded symbolically for 2, 3 and 4-Bar kinematic systems, Table 8.2 is obtained. (Terms containing diagonal elements are shown in italics).

Coefficient	2-Bar System		3-Bar System		4-Bar System	
p_0	1	λ^2	1	λ^3	1	λ^4
p_1	$-d_{11}$	λ	$-d_{11}$	λ^2	$-d_{11}$	λ^3
	$-d_{22}$		$-d_{22}$		$-d_{22}$	
			$-d_{33}$		$-d_{33}$	
					$-d_{44}$	
p_2	$d_{11}d_{22}$	constant	$d_{11}d_{33}$	λ	$d_{11}d_{33}$	λ^2
	$-d_{12}d_{21}$	terms	$d_{22}d_{33}$		$d_{11}d_{44}$	
			$d_{11}d_{22}$		$d_{33}d_{44}$	
			$-d_{31}d_{13}$		$d_{22}d_{44}$	
			$-d_{21}d_{12}$		$d_{22}d_{33}$	
			$-d_{32}d_{23}$		$d_{11}d_{22}$	
					$-d_{34}d_{43}$	

					$-d_{32}.d_{23}$	
					$-d_{42}d_{24}$	
					$-d_{21}d_{12}$	
					$-d_{31}d_{13}$	
					$-d_{41}d_{14}$	
p_3			$-d_{11}(d_{22}d_{33})$	constant	$-d_{22}d_{33}d_{44}$	λ
			$d_{22}\ (d_{31}d_{13})$	terms	$-d_{11}d_{33}d_{44}$	
			$d_{11}\ (d_{32}d_{23})$		$-d_{11}d_{22}d_{44}$	
			$d_{33}\ (d_{21}d_{12})$		$-d_{11}d_{22}d_{33}$	
			$-d_{12}d_{23}d_{31}$		$d_{11}(d_{34}d_{43})$	
			$-d_{13}d_{21}d_{32}$		$d_{22}(d_{34}d_{43})$	
					$d_{44}(d_{32}d_{23})$	
					$d_{32}d_{24}d_{43}$	
					$d_{42}d_{23}d_{34}$	
					$d_{42}d_{24}d_{33}$	
					$d_{11}(d_{32}d_{23})$	
					$d_{11}(d_{42}d_{24})$	
					$d_{44}(d_{21}d_{12})$	
					$d_{33}(d_{21}d_{12})$	
					$d_{21}d_{32}d_{13}$	
					$d_{21}d_{42}d_{14}$	
					$d_{31}d_{12}d_{23}$	
					$d_{44}(d_{31}d_{13})$	
					$d_{31}d_{14}d_{43}$	
					$d_{22}(d_{31}d_{13})$	
					$d_{41}d_{12}d_{24}$	
					$d_{41}d_{13}d_{34}$	
					$d_{33}(d_{41}d_{14})$	
					$d_{22}(d_{41}d_{14})$	
P_4					$d_{11}d_{22}d_{33}d_{44}$	constant
					$d_{21}d_{12}d_{34}d_{43}$	terms
					$d_{31}d_{42}d_{13}d_{24}$	
					$d_{41}d_{32}d_{14}d_{23}$	
					$-d_{11}d_{22}(d_{34}d_{43})$	
					$-d_{11}d_{44}(d_{32}d_{23})$	
					$-d_{11}(d_{32}d_{24}d_{43})$	
					$-d_{11}(d_{42}d_{23}d_{34})$	
					$-d_{11}d_{33}(d_{42}d_{24})$	
					$-d_{33}d_{44}(d_{21}d_{12})$	
					$-d_{44}(d_{21}d_{32}d_{13})$	
					$-d_{21}d_{32}d_{14}d_{43}$	
					$-d_{21}d_{42}d_{13}d_{34}$	
					$-d_{33}(d_{21}d_{42}d_{14})$	
					$-d_{44}(d_{31}d_{12}d_{23})$	
					$-d_{31}d_{12}d_{24}d_{43}$	
					$-d_{22}d_{44}(d_{31}d_{13})$	
					$-d_{22}(d_{31}d_{14}d_{43})$	
					$-d_{31}d_{42}d_{14}d_{23}$	
					$-d_{41}d_{12}d_{23}d_{34}$	
					$-d_{33}(d_{41}d_{12}d_{24})$	
					$-d_{22}(d_{41}d_{13}d_{34})$	
					$-d_{22}d_{33}(d_{41}d_{14})$	
					$-d_{41}d_{32}d_{13}d_{24}$	

Table 8.2: Comparison of the Symbolic Expansions of the Determinants formed from 2, 3 and 4-Bar Kinematic System Constraints Matrices

Substituting numerical values for typical, revolute jointed, systems into Table 8.2, and applying the notation defined in 8.1.1, Table 8.3 is obtained.

Coeff.	2-Bar System		3-Bar Chain		3-Bar Closed Loop		4-Bar Chain		4-Bar Closed Loop	
p_0	1	λ^2	1	λ^3	1	λ^3	1	λ^4	1	λ^4
p_1	-6		-6		-6		-6		-6	
	-6		-6		-6		-6		-6	
	= -12	λ	-6		-6		-6		-6	
			= - 18	λ^2	= - 18	λ^2	-6		-6	
							= -24	λ^3	= -24	λ^3
p_2	36		36		36		36		36	
	-25	**2-loop**	36		36		36		36	
	= 11		36		36		36		36	
			0		**-25**	**2-loop**	36		36	
			-25	**2-loop**	**-25**	**2-loop**	36		36	
			-25	**2-loop**	**-25**	**2-loop**	36		36	
			= 58	λ	= 33	λ	-25	**2-loop**	-25	**2-loop**
							-25	**2-loop**	-25	**2-loop**
							0		0	
							-25	**2-loop**	-25	**2-loop**
							0		0	
							0		-25	**2-loop**
							= 141	λ^2	= 116	λ^2
p_3			-216		-216		-216		-216	
			0		150	1.2-loop	-216		-216	
			150	1.2-loop	150	1.2-loop	-216		-216	
			150	1.2-loop	150	1.2-loop	-216		-216	
			0		**-125**	**3-loop**	150	1.2-loop	150	1.2-loop
			0		**-125**	**3-loop**	150	1.2-loop	150	1.2-loop
			= 84		= -16		150	1.2-loop	150	1.2-loop
							0		0	
							0		0	
							0		0	
							150	1.2-loop	150	1.2-loop
							0		0	
							150	1.2-loop	150	1.2-loop
							150	1.2-loop	150	1.2-loop
							0		0	
							0		0	
							0		0	
							0		0	
							0		0	
							0		0	
							0		0	
							0		0	
							0		150	1.2-loop
							0		150	1.2-loop
							= 36	λ	= 336	λ
p_4							1296		1296	
							625	2.2-loop	625	2.2-loop
							0		0	
							0		625	2.2-loop
							-900	1.2-loop	-900	1.2-loop

							-900	1.2-loop	-900	1.2-loop
							0		0	
							0		0	
							0		0	
							-900	1.2-loop	-900	1.2-loop
							0		0	
							0		**-625**	**4-loop**
							0		0	
							0		0	
							0		0	
							0		0	
							0		0	
							0		0	
							0		0	
							0		**-625**	**4-loop**
							0		0	
							0		0	
							0		-900	1.2-loop
							0		0	
							= -779		= - 2304	

Table 8.3: Numerical Values for the Determinant Terms in Table 8.2for Typical, Simple 2, 3 and 4-Bar Kinematic Systems

For the five sample systems in Table 8.3, the constraints matrix characteristic polynomials are known and are shown in Table 8.4 (see Appendix C, Appendix D, and Appendix E):

System Description	**Characteristic Polynomial, based on Constraints Matrix,**	**Eigenvalues**
2-Bar chain	$\lambda^2 - 12\lambda + 11$	$\lambda_1 = 1, \lambda_2 = 11$
3-Bar chain	$\lambda^3 - 18\lambda^2 + 58\lambda + 84$	$\lambda_1 = 13.071, \lambda_2 = 6, \lambda_3 = -1.071$
3-Bar closed loop	$\lambda^3 - 18\lambda^2 + 33\lambda - 16$	$\lambda_1 = 1, \lambda_2 = 1, \lambda_3 = 16$
4-Bar chain	$\lambda^4 - 24\lambda^3 + 141\lambda^2 + 36\lambda - 779$	$\lambda_1 = 9.09, \lambda_2 = 14.09, \lambda_3 = 2.91, \lambda_4 = -2.09$
4-Bar closed loop	$\lambda^4 - 24\lambda^3 + 116\lambda^2 + 336\lambda - 2304$	$\lambda_1 = 6, \lambda_2 = 16, \lambda_3 = 6, \lambda_4 = -4$

Table 8.4: Characteristic Polynomials and Eigenvalues for the Sample Systems in Table 8.3

The numerical values of the characteristic polynomial coefficients can be derived from the values in Table 8.3. Using an inspection method similar to that employed by Yan and Hall [90], the process is shown in Table 8.5.

$p_0 =$		=	$+\{({}^nC_r)(\text{max. constraints})^r\} = 1$
$p_1 =$	$-\Sigma\,\lambda_i$	=	$-\{({}^nC_r)(\text{max. constraints})^r\} = -$ trace of Constraints Matrix
$p_2 =$	$\Sigma\,\lambda_i\lambda_j$ $i \neq j$	=	$+\{({}^nC_r)(\text{max. constraints})^r\}$ $-$ {(no. of 2-loops)(product of constraints on the edges of each 2-loop)}
$p_3 =$	$-\Sigma$ $\lambda_i\lambda_j\lambda_k$ $i \neq j \neq k$	=	$-\{({}^nC_r)(\text{max. constraints})^r\}$ $+$ {(n-2)(no. of 2-loops)(max. constraints)(product of constraints on 2-loop edges)} $-$ 2{(no. of 3-loops)(product of constraints on edges of each 3-loop)
$p_4 =$	Σ $\lambda_i\lambda_j\lambda_k\lambda_l$ $i \neq j \neq k$ $\neq l$	=	$+\{({}^nC_r)(\text{max. constraints})^r\}$ $-$ {(no. of 2-loops)(max. constraints)^2(product of constraints on 2-loop edges)} $+$ {(no. of 2.2 loops)(product of constraints on edges of both 2-loops)} $+$ 2{(no. of 3-loops)(max. constraints)(product of constraints on 3-loop edges)} $-$ 2{(no. of 4-loops)(product of constraints on 4-loop edges)}
....	.....	.. .	.....
$p_n =$	$-\lambda_1\lambda_2.....\lambda_n$	=	$\lvert\mathbf{C}\rvert$ ie the determinant of the matrix **C**.

Table 8.5: Characteristic Polynomial Coefficients by Inspection

In Table 8.5, ${}^nC_r = \binom{n}{r} = \dfrac{n!}{r!(n-r)!}$, the number of unordered combinations of r items selected from a set of n items ('n choose r', ${}^nC_r$), and by convention, 0! = 1, eg Anderson [4]. Here n represents the number of vertices (ie links) in the system, r is the characteristic polynomial coefficient subscript, and $\lambda_1, \lambda_2, ..., \lambda_n$ are the eigenvalues of the characteristic polynomial for the system under consideration.

As presented in Table 8.5, the final (p_n) coefficient of a characteristic polynomial is the product of all of the eigenvalues. Thus, the subscript, r, of the coefficients can never be greater than the order of the matrix, n. Therefore, there is no need to evaluate coefficients for r > n, since the resulting 'coefficient' will have no meaning. Given this fact, the results of Table 8.5 can be applied to carry out a numerical validity check against the systems evaluated in Table 8.3 as follows (where N_2 is the number of '2-loops', N_{22} is the number of joint '2.2-loops', N_3 is the number of joint '3-loops', and N_4 is the number of joint '4-loops' in the symbolically expressed characteristic polynomial):

For a 2-Bar system (ie n = 2, and ${}^nC_r = {}^2C_r$):

$N_2 = 1$

$p_0 = (1)(6^0) = 1$

$p_1 = -(2)(6^1) = -12$

$p_2 = +\{(1)(6^2)\} - \{(N_2)(5^2)\} = 36 - 25 = 11$

It can also be shown that the numerical values of these coefficients can be verified from the eigenvalues in Table 8.4 thus:

$p_1 = -(\lambda_1 + \lambda_2) = -(1 + 11) = -12$

$p_2 = (\lambda_1 \times \lambda_2) = (11 \times 1) = 11$

For a 3-Bar closed system (ie n = 3, and ${}^{n}C_r = {}^{3}C_r$):

$N_2 = 3, N_3 = 1$

$p_0 = (1)(6^0) = 1$

$p_1 = -(3)(6^1) = -18$

$p_2 = +\{(3)(6^2)\} - \{(N_2)(5^2)\} = 108 - 75 = 33$

$p_3 = -\{(1)(6^3)\} + \{(1)(N_2)(6)(5^2)\} - 2\{(N_3)(5^3)\} = -216 + 450 - 250 = -16$

Here it can be shown that the numerical values of the higher coefficients can be verified from the eigenvalues in Table 8.4 as follows:

$p_2 = (\lambda_1\lambda_2 + \lambda_2\lambda_3 + \lambda_1\lambda_3) = (1 \times 1) + (1 \times 16) + (1 \times 16) = 33$

$p_3 = -\lambda_1\lambda_2\lambda_3 = -16$

For a 4-Bar closed system (ie n = 4, and ${}^{n}C_r = {}^{4}C_r$):

$N_2 = 4, N_{22} = 2, N_3 = 0, N_4 = 1$

$p_0 = (1)(6^0) = 1$

$p_1 = -(4)(6^1) = -24$

$p_2 = +\{(6)(6^2)\} - \{(N_2)(5^2)\} = 216 - 100 = 116$

$p_3 = -\{(4)(6^3)\} + \{(2)(N_2)(6)(5^2)\} - 2\{(0)(5^3)\} = 336$

$p_4 = +\{(1)(6^4)\} - \{2(N_2)\}(6^2)(5^2) + \{(N_{22})(5^2)(5^2) - (2)(N_4)(5^4)\}$

$= 1296 - 3600 + 1250 - 1250 = -2304$

Again, it can be shown that the numerical values of the higher coefficients can be verified from the eigenvalues in Table 8.4 as follows:

$p_2 = (\lambda_1\lambda_2 + \lambda_2\lambda_3 + \lambda_1\lambda_3 + \lambda_2\lambda_4 + \lambda_1\lambda_4 + \lambda_3\lambda_4) = (96 + 96 + 36 - 64 - 24 - 24) = 116$

$p_3 = -(\lambda_1\lambda_2\lambda_3 + \lambda_2\lambda_3\lambda_4 + \lambda_1\lambda_3\lambda_4 + \lambda_1\lambda_2\lambda_4 = -(576 - 384 - 144 - 384) = 336$

$p_4 = \lambda_1\lambda_2\lambda_3\lambda_4 = (6 \times 16 \times 6 \times (-4) = -2304$

Therefore, on the basis of the foregoing three examples, it is considered that the evaluation of characteristic polynomial coefficients based on the coefficient individual constituent terms (Table 8.3), and as derived from the characteristic polynomial eigenvalues (Table 8.4), are consistent and compatible with one another.

Additionally, a 'prediction' for a system outside Table 8.3 can be carried out in order to further validate the method:

For a 4-Bar looped system (ie n = 4, and ${}^{n}C_{r} = {}^{4}C_{r}$):

$N_2 = 4, N_{22} = 1, N_3 = 1, N_4 = 0$

$p_0 = (1)(6^0) = 1$

$p_1 = -(4)(6^1) = -24$

$p_2 = +\{(6)(6^2)\} - \{4(N_2)(5^2)\} = 216 - 100 = 116$

$p_3 = -\{(4)(6^3)\} + \{(2)(N_2)(6)(5^2)\} - 2\{(N_3)(5^3)\} = -864 + 1200 - 1114 = -86$

$p_4 = +\{(1)(6^4)\} + \{(1)(N_2)(6^2)(5^2) + (N_{22})(5^2)(5^2) + \{(2)(N_3)(5^3)(6) - (2)(0)(625)$

$= 1921 - 3600 + 1500 = -179$

The values obtained are consistent with the known eigenvalues of the system evaluated, and therefore it is considered that reasonable confidence can be assumed that the method is sound, and that direct comparisons can be drawn with work carried out to derive a 'coefficients by inspection' method for Characteristic Polynomials derived from the Adjacency Matrix [90]. It is therefore considered justifiable to continue to investigate fault behaviour based on the methods developed so far.

8.3 Robustness Indications in Coefficients by Inspection

It has been shown in the previous section that it is possible to develop for the Constraints Matrix, **C**, a direct equivalent of the 'Coefficients by Inspection' rule developed for the Adjacency Matrix, **A**, by Yan and Hall [90]. In order to proceed with any system selection / comparison program on a sound basis, it is necessary to be clear how system robustness can be illustrated and represented using the revised 'Coefficients by Inspection' rule that has been identified for the Constraints Matrix, **C**, (Table 8.4).

8.3.1 Fault Digraphs and Transition Paths of Sample Systems

8.3.1.1 Fault State Digraph

A 'fault path' describing the evolution of a kinematic system from its nominal operational state through several kinematic fault states, can be represented as a digraph of fault states. These fault states fall into the following different categories:

- ***nominal state*** – the unfailed, starting, condition

- ***substitute operational state*** – a fault condition which, nonetheless leaves the system with some degree of its original mobility
- ***non-operational state*** – a fault condition in which the system cannot operate. This can be considered to have a subdivision, here referred to as an ***intermediate enabling state***, which is considered to be a state into which a system can be placed by choice, which can be used as an intermediate stage in moving to an alternative fault condition.
- ***non-operational absorbing state*** – a terminal condition in which the system cannot operate, and from which there is no recovery – ie failure.

This progression from nominal state, through fault states to one or more substitute operational state(s) or a final failure may be regarded as a Markov Chain, [88, 40], and any non-operational states as 'absorbing states'.

The application of Markov chain theory is not explored here, but is considered worthy of note. This theory might allow the definition of the transition matrix for any system and its corresponding family of fault configurations from the fault state digraph. The transition matrix embeds the probabilities of transition from one state to another, and so has a direct relationship to the principles of entropy definition [60] as previously discussed in Chapter 7. This might allow an alternative means of establishing the entropy of each of the fault states, and, potentially could be used to develop the definition of fault classes (which will be discussed later in this chapter).

8.3.1.2 Kinematic Joint Failure Mode Assumptions

The following system analyses are carried out in 2D, because this simplifies the failure modes and makes it easier to visualise what is happening.

Where a kinematic joint type changes its nature through fault action, the notation used is that of the original joint type. In 2D, it is assumed that any undriven, 1 degree of freedom, 2-loop can have four states – nominal plus three fault conditions:

- Failed locked (one less degree of freedom)
- Failed separated (two additional degrees of freedom)
- Failed with one additional degree of freedom

8.3.1.3 Sample Systems

Several sample systems are considered, as follows, and investigated in detail in subsequent sections:

- Undriven Dyad
- Robust Undriven Dyad ('LH Anchored')
- Robust Undriven Dyad ('RH Anchored')
- Pantograph Leg Mechanism [38]
- Five-Bar Mechanism based on Attila leg [5]

8.3.1.3.1 Undriven Dyad

Consider the behaviour of an undriven dyad when it undergoes transition through various fault stages. The system has four possible system states, and four transitions, as identified in Table 8.6.

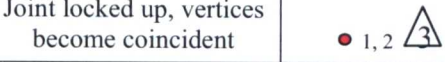
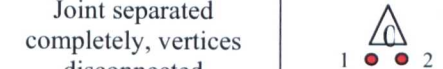
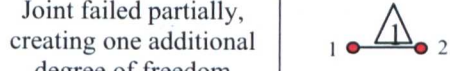
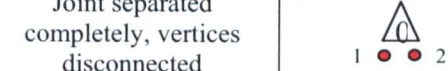
Initial State	Initial State Description	Initial State Diagram	Second State	Second State Description	Second State Diagram
S1	Nominal – fully functional		S2	Joint locked up, vertices become coincident	
S1	Nominal – fully functional		S3	Joint separated completely, vertices disconnected	
S1	Nominal – fully functional		S4	Joint failed partially, creating one additional degree of freedom	
S4	Joint failed partially, creating one additional dof		S3	Joint separated completely, vertices disconnected	

Table 8.6: State Table for Undriven Dyad

From the above, and applying the principles outlined in 8.3.1.1, the corresponding fault state digraph can be derived:

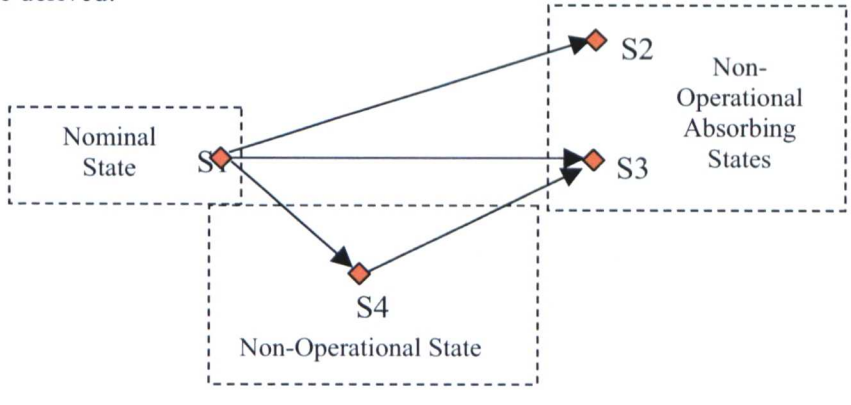
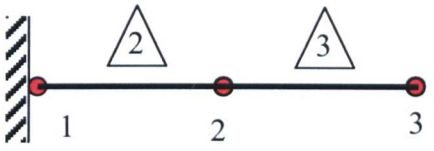


Figure 8.2:Fault State Digraph for Undriven Dyad

8.3.1.4 Robust Undriven Dyad ('LH Anchored')

This system (shown schematically below) is a *three*-bar linkage, with one joint locked so that the linkage acts as a dyad, but with the capability to unlock the 'standby' or 'latent' joint to regain lost functionality. The system is anchored at the end nearest the active joint.



The system has 7 system states, and 8 transitions, as shown in Table 8.7:

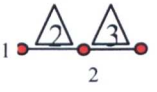
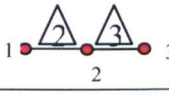
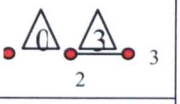
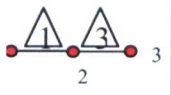
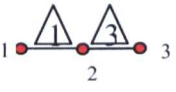
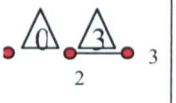
Initial State	Initial State Description	Initial State Diagram	Second State	Second State Description	Second State Diagram
S1	Nominal – fully functional		S2	Joint between 1 and 2 locks up, 1 and 2 become coincident.	
S2	Joint between 1 and 2 locks up, 1 and 2 become coincident.		S5	Functionality restored by release of joint between 2 and 3.	
S1	Nominal – fully functional		S3	Joint between 1 and 2 becomes separated. Functionality cannot be restored.	
S1	Nominal – fully functional		S4	Joint between 1 and 2 fails partially, creating one additional degree of freedom	
S4	Joint between 1 and 2 fails partially, creating one additional degree of freedom		S2	Functionality unlikely to be restored. Locking of joint between 1 and 2 may freeze both dofs (S2), & allow release of joint between 2 and 3 (S6).	
S2	Joint between 1 and 2 locks up, 1 and 2 become coincident.		S3	Joint between 1 and 2 becomes separated. Functionality cannot be restored.	
S5	Functionality restored by release of joint between 2 and 3.		S6	Joint between 2 and 3 fails giving an additional dof	
S6	Joint between 2 and 3 fails giving an additional dof		S7	Joint between 2 and 3 separates completely	

Table 8.7: State Table for Robust Dyad 'LH Anchored'

From the above, the corresponding fault state digraph can be derived:

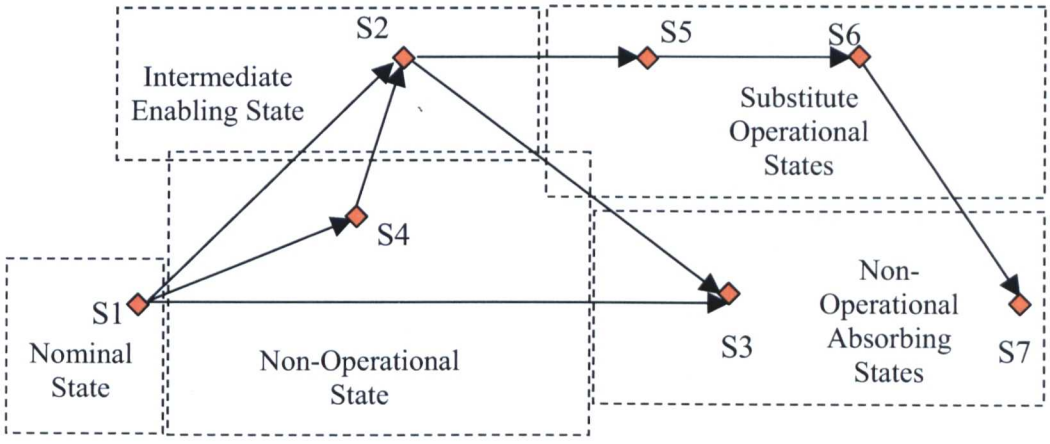
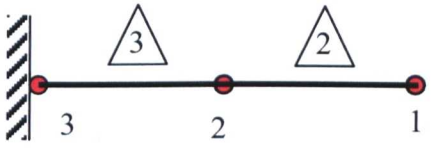


Figure 8.3:Fault State Digraph for Robust Dyad 'LH Anchored'

8.3.1.5 Robust Undriven Dyad ('RH Anchored')

This system (shown schematically below) is a *three*-bar linkage, with one joint locked so that the linkage acts as a dyad, but with the capability to unlock the 'standby' joint to regain lost functionality. The system is anchored at the end furthest from the active joint.



The system has 10 system states, and 11 transitions, as shown in Table 8.8:

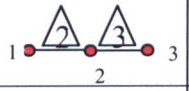
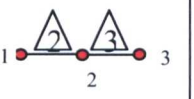
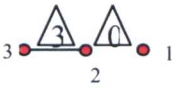
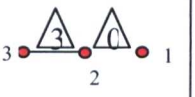
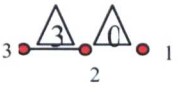
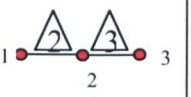
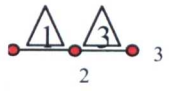
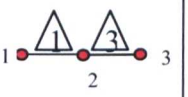
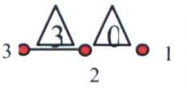
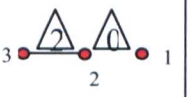
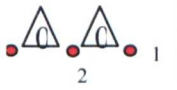
Initial State	Initial State Description	Initial State Diagram	Second State	Second State Description	Second State Diagram
S1	Nominal – fully functional		S2	Joint between 1 and 2 locks up, 1 and 2 become coincident.	
S2	Joint between 1 and 2 locks up, 1 and 2 become coincident.		S5	Functionality restored by release of joint between 2 and 3.	
S1	Nominal – fully functional		S3	Joint between 1 and 2 becomes separated. Functionality can be restored.	
S3	Joint between 1 and 2 becomes separated. Functionality can be restored.		S6	Functionality restored by release of joint between 2 and 3.	
S1	Nominal – fully functional		S4	Joint between 1 and 2 fails partially, creating one additional degree of freedom	
S4	Joint between 1 and 2 fails partially, creating one additional degree of freedom		S2	Functionality unlikely to be restored. Locking of joint between 1 and 2 may freeze both dofs (S2), & allow release of joint between 2 and 3 (S6).	
S2	Joint between 1 and 2 locks up, 1 and 2 become coincident.		S3	Joint between 1 and 2 becomes separated. Functionality can be restored.	
S5	Functionality restored by release of joint between 2 and 3.		S7	Joint between 2 and 3 fails giving an additional dof	
S7	Joint between 2 and 3 fails giving an additional dof		S8	Joint between 2 and 3 separates completely. System leaves anchorage.	
S6	Functionality restored by release of joint between 2 and 3.		S9	Joint between 2 and 3 fails giving an additional dof	
S9	Joint between 2 and 3 fails giving an additional dof		S10	Joint between 2 and 3 separates completely	

Table 8.8: State Table for Robust Dyad 'RH Anchored'

From the above, the corresponding digraph, can be derived:

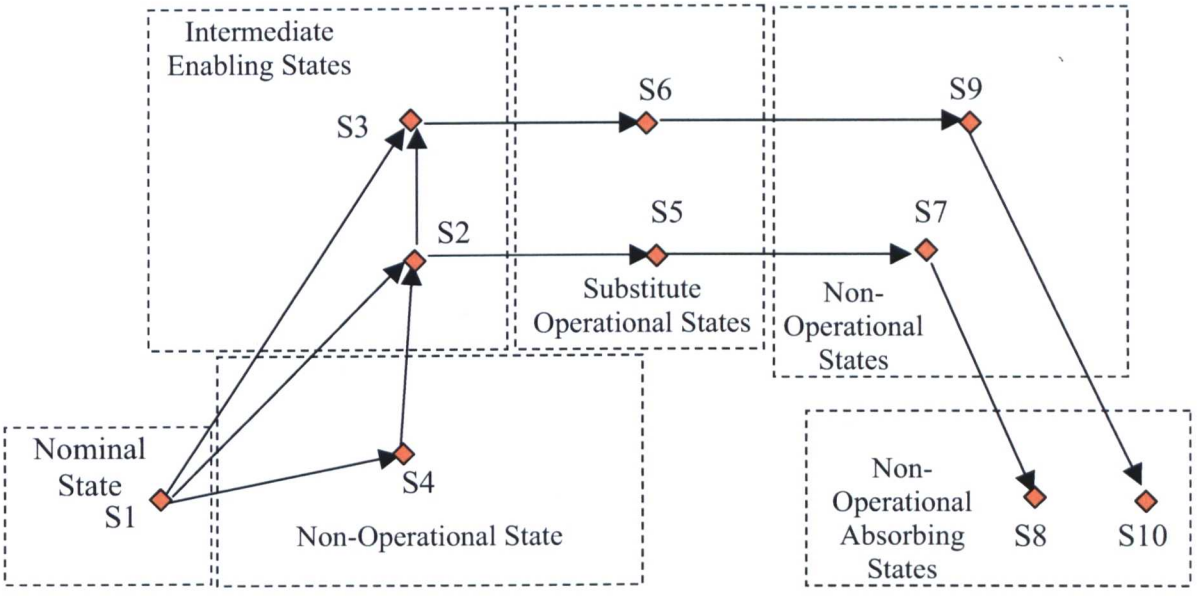


Figure 8.4:Fault State Digraph for Robust Dyad 'RH Anchored'

8.3.1.6 Pantograph Leg Mechanism

If the actuator mechanism is ignored, this system (shown in Figure 8.5) is a *four*-bar 'pantograph' linkage, comprising four revolute joints such that the four-bar parallelogram can be employed as part of a leg mechanism to produce a vertical 'stepping' motion.

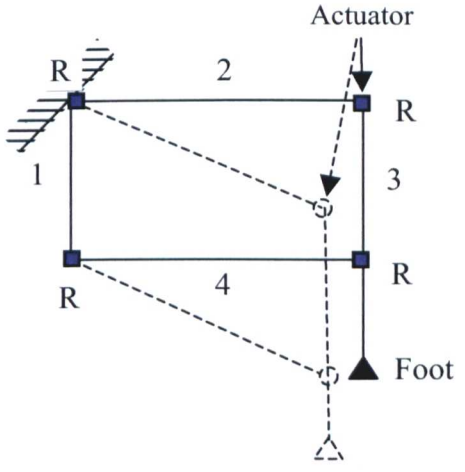


Figure 8.5:Schematic Representation of Pantograph Mechanism

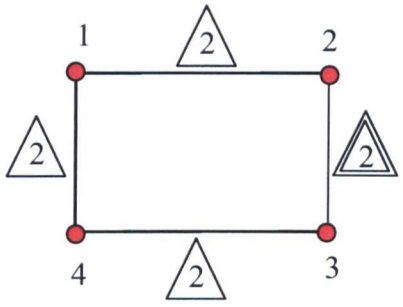
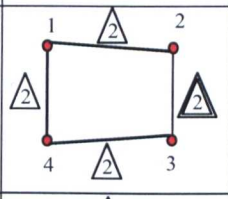
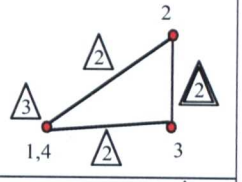
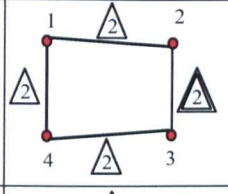
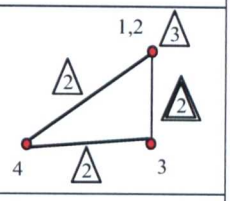
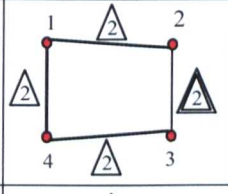
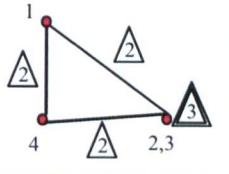
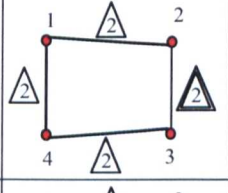
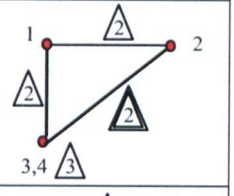
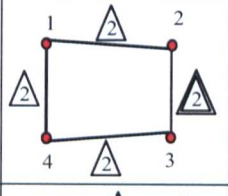
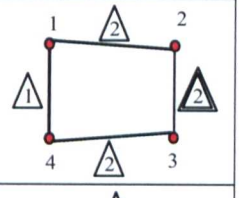
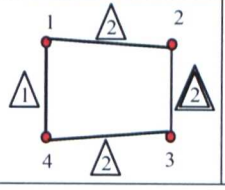
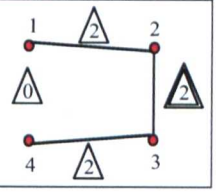
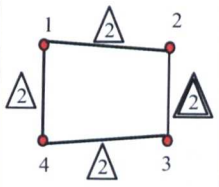
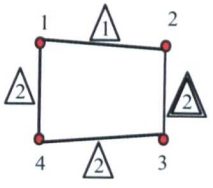
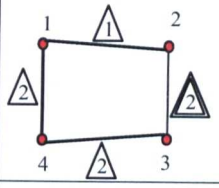
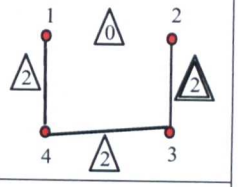
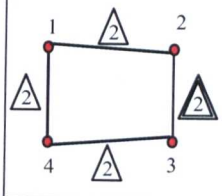
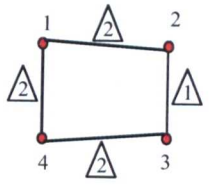
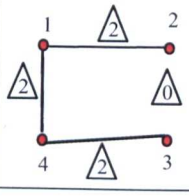
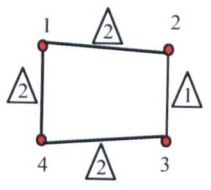
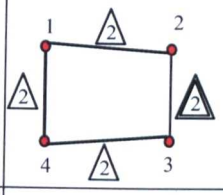
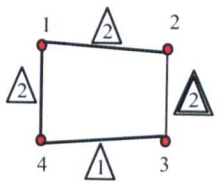
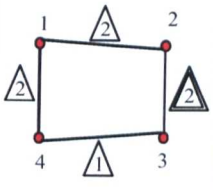
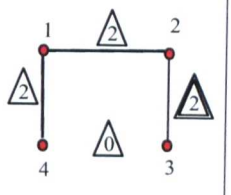
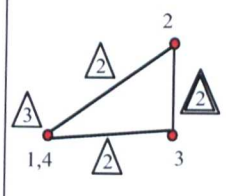
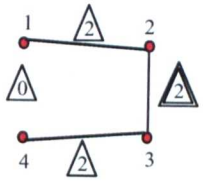
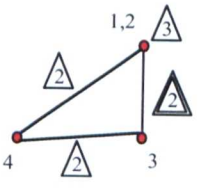
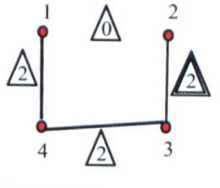


Figure 8.6:Interchange Graph Representation of Pantograph Mechanism

The system has 18 system states, and 22 transitions, as shown in Table 8.9:

Initial State	Initial State Description	Initial State Diagram	Second State	Second State Description	Second State Diagram
S1	Nominal – fully functional		S2	Revolute joint between 1 and 4 locks up. 1 and 4 become coincident. Limited functionality remains.	
S1	Nominal – fully functional		S3	Revolute joint between 1 and 2 locks up. 1 and 2 become coincident. This state is unrecoverable.	
S1	Nominal – fully functional		S4	Revolute joint between 2 and 3 locks up. 2 and 3 become coincident. Limited functionality remains.	
S1	Nominal – fully functional		S5	Revolute joint between 3 and 4 locks up. 3 and 4 become coincident. Limited functionality remains.	
S1	Nominal – fully functional		S14	Revolute joint between 4 and 1 gains 1 dof.	
S14	Revolute joint between 4 and 1 gains 1 dof.		S6	Revolute joint between 4 and 1 becomes separated. This state is unrecoverable, because actuator comes off.	

S1	Nominal – fully functional		S15	Revolute joint between 1 and 2 gains 1 dof.	
S15	Revolute joint between 1 and 2 gains 1 dof.		S7	Revolute joint between 1 and 2 becomes separated. This state is unrecoverable, because system leaves wall.	
S1	Nominal – fully functional		S16	Revolute joint between 2 and 3 gains 1 dof.	
S16	Revolute joint between 2 and 3 gains 1 dof.		S8	Revolute joint between 2 and 3 becomes separated.	
S1	Nominal – fully functional		S17	Revolute joint between 3 and 4 gains 1 dof.	
S17	Revolute joint between 3 and 4 gains 1 dof.		S9	Revolute joint between 3 and 4 becomes separated.	
S2	Revolute joint between 1 and 4 locks up. 1 and 4 become coincident. Limited functionality remains.		S6	Revolute joint between 4 and 1 becomes separated. This state is unrecoverable.	
S3	Revolute joint between 1 and 2 locks up. 1 and 2 become coincident. This state is unrecoverable.		S7	Revolute joint between 1 and 2 becomes separated. This state is unrecoverable.	

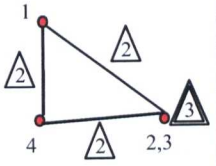
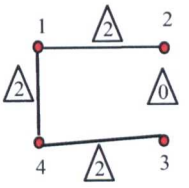
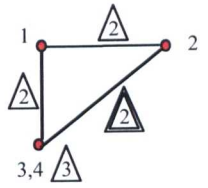
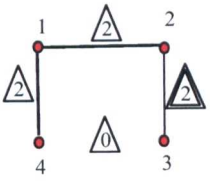
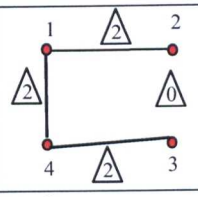
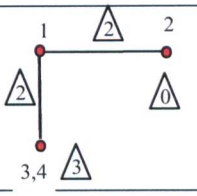
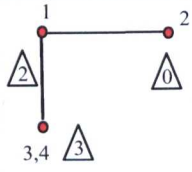
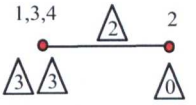
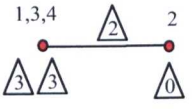
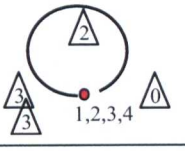
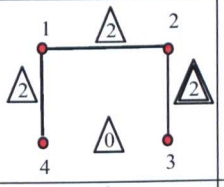
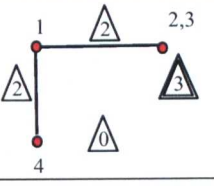
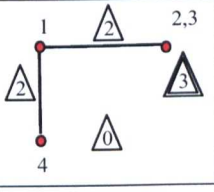
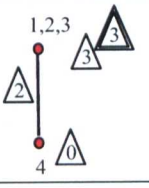
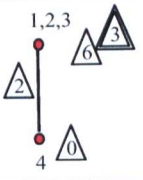
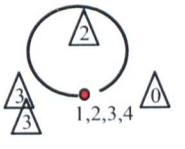
S4	Revolute joint between 2 and 3 locks up. 2 and 3 become coincident. Limited functionality remains.		S8	Revolute joint between 2 and 3 becomes separated.	
S5	Revolute joint between 3 and 4 locks up. 3 and 4 become coincident. Limited functionality remains.		S9	Revolute joint between 3 and 4 becomes separated.	
S8	Revolute joint between 2 and 3 becomes separated.		S10	Lock joint between 4 and 3 as intermediate enabling state.	
S10	Lock joint between 4 and 3 as intermediate enabling state.		S18	Lock joint between 1 and 4 to regain limited functionality.	
S18	Lock joint between 1 and 4 to regain limited functionality		S11	Joining 2 to 4 provides additional fault tolerance with limited functionality.	
S9	Revolute joint between 3 and 4 becomes separated.		S12	Lock joint between 2 and 3 as intermediate enabling state.	
S12	Lock joint between 2 and 3 as intermediate enabling state.		S13	Lock joint between 1 and 2 to regain limited functionality.	
S13	Lock joint between 1 and 2 to regain limited functionality		S11	Joining 4 to 2 provides additional fault tolerance with limited functionality.	

Table 8.9: State Table for Pantograph Mechanism

From the above, the corresponding digraph can be derived:

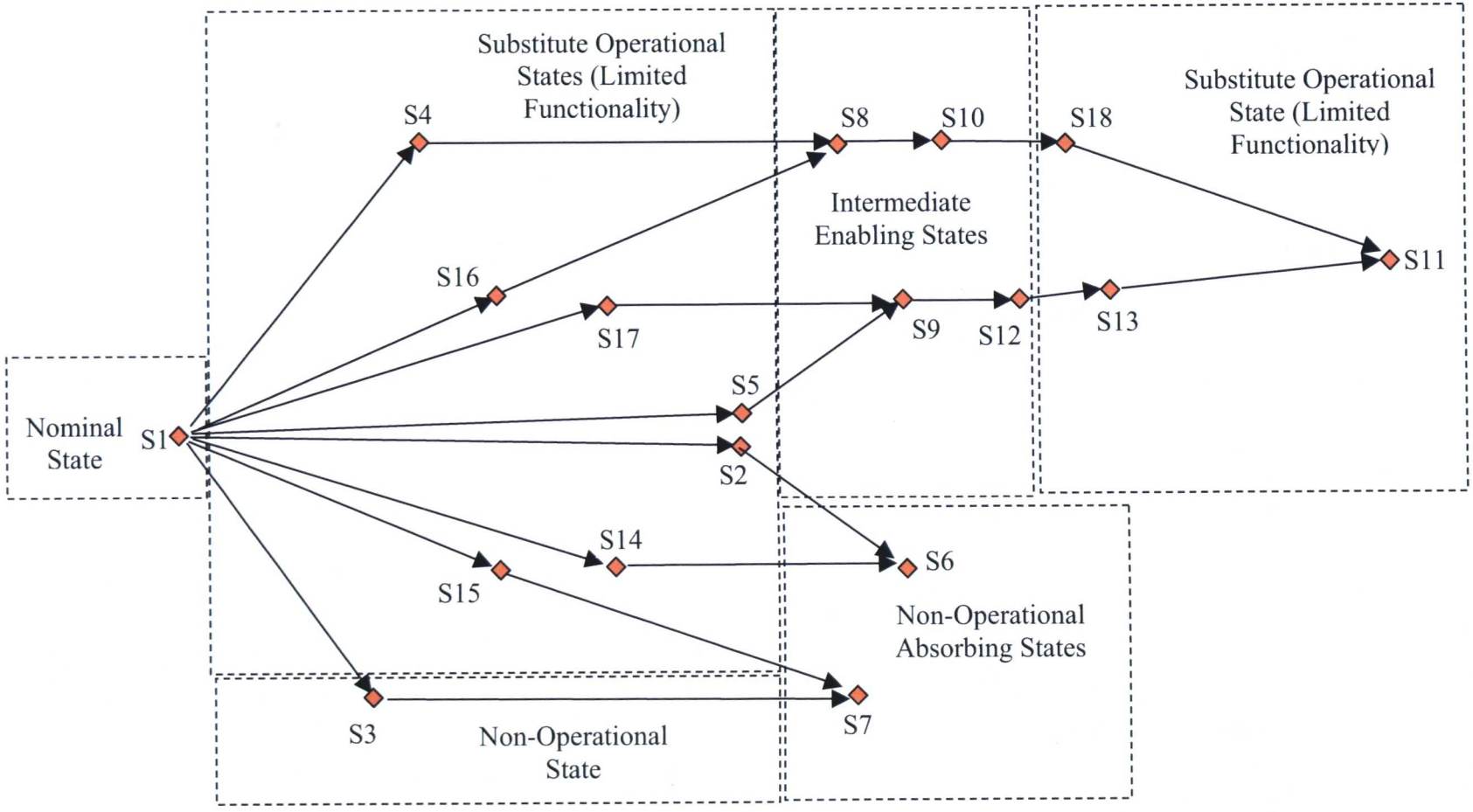


Figure 8.7:Digraph for Pantograph Mechanism

8.3.1.7 Kinematic System Based on Attila Leg Mechanism

This system is a *five*-bar linkage, comprising four revolute joints and one prismatic joint such that the five-bar polygon can be reshaped to produce a stepping motion which is a combination of outward swing and vertical movement. (In the complete robot, this motion is combined with a fore-and-aft swing to produce the complete step).

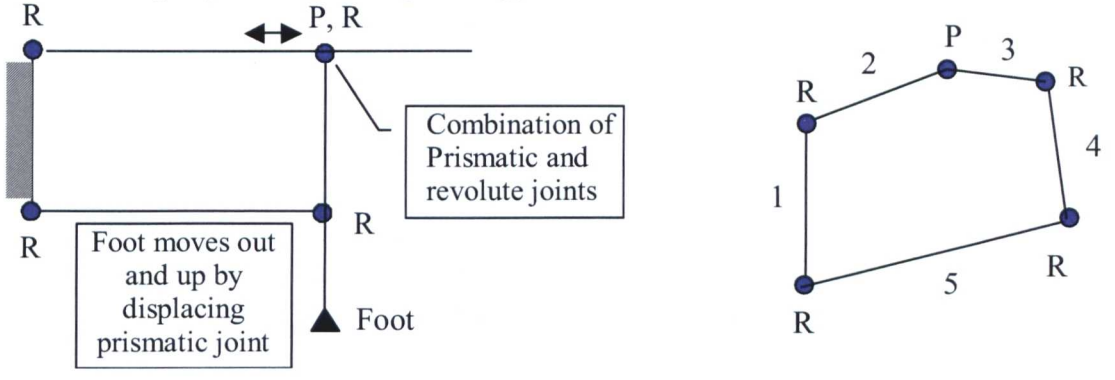


Figure 8.8: Schematic Representation of System based on Attila Leg

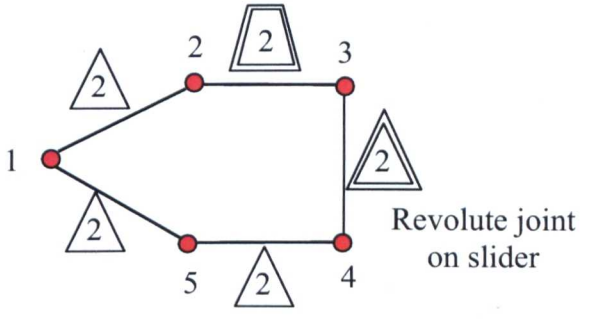
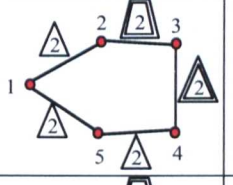
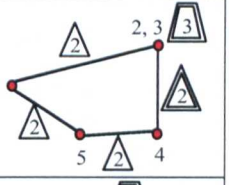
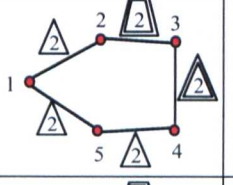
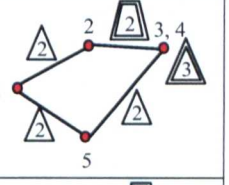
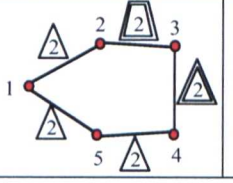
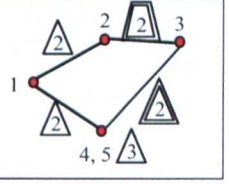
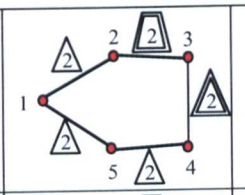
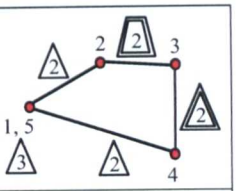
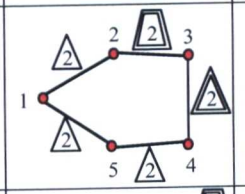
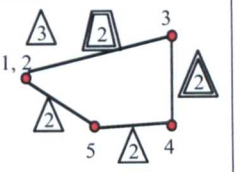
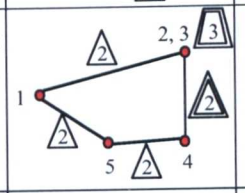
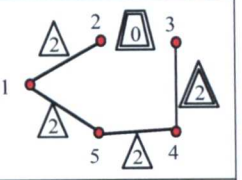
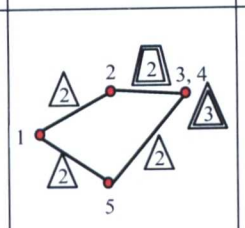
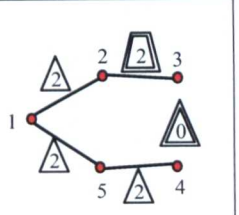
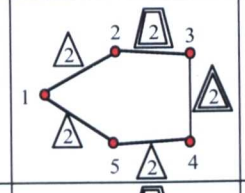
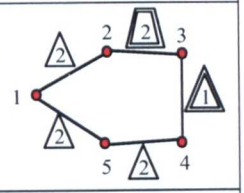
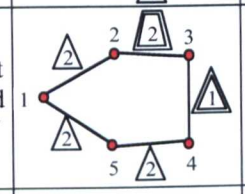
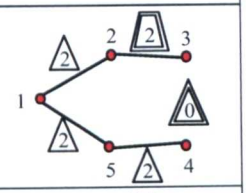
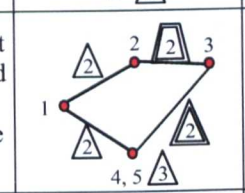
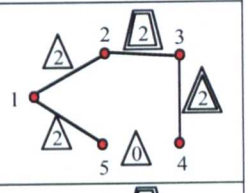
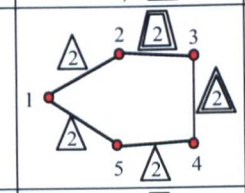
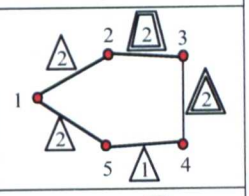
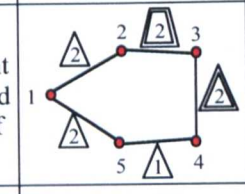
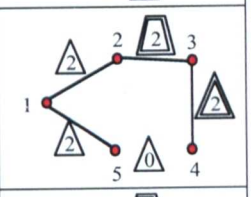
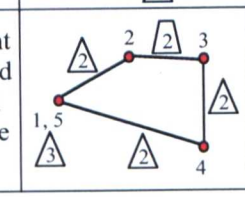
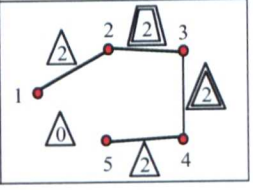


Figure 8.9:Interchange Graph Representation of System Based on Attila Leg

The system has 19 system states, and 22 transitions, as shown in Table 8.10:

Initial State	Initial State Description	Initial State Diagram	Second State	Second State Description	Second State Diagram
S1	Nominal – fully functional		S2	Prismatic joint between 2 and 3 locks up, 2 and 3 become coincident. 4 loop with 'loss of edge'	
S1	Nominal – fully functional		S3	Revolute joint on slider (between 3 and 4) locks up, 3 and 4 become coincident 4 loop with 'loss of edge'	
S1	Nominal – fully functional		S4	Revolute joint between 4 and 5 locks up, 4 and 5 become coincident. 4 loop with 'loss of edge'	

S1	Nominal – fully functional		S5	Revolute joint between 1 and 5 locks up, 1 and 5 become coincident. 4 loop with 'loss of edge'	
S1	Nominal – fully functional		S6	Revolute joint between 1 and 2 locks up, 1 and 2 become coincident. 4 loop with 'loss of edge'	
S2	Prismatic joint between 2 and 3 locks up, 2 and 3 become coincident.		S7	Prismatic joint fails separated. No recovery possible. 5 bar chain	
S3	Revolute joint on slider (between 3 and 4) locks up, 3 and 4 become coincident		S8	Revolute joint between 3 and 4 fails separated. 5 bar chain	
S1	Nominal – fully functional		S9	Revolute joint between 3 and 4 gains 1 dof. 5 loop 'without loss of edge'	
S9	Revolute joint between 3 and 4 gains 1 dof		S8	Revolute joint between 3 and 4 fails separated. 5 bar chain	
S4	Revolute joint between 4 and 5 locks up, 4 and 5 become coincident		S10	Revolute joint between 4 and 5 fails separated. 5 bar chain	
S1	Nominal – fully functional		S11	Revolute joint between 4 and 5 gains 1 dof. 5 loop 'without loss of edge'	
S11	Revolute joint between 4 and 5 gains 1 dof		S10	Revolute joint between 4 and 5 fails separated. 5 bar chain	
S5	Revolute joint between 1 and 5 locks up, 1 and 5 become coincident		S12	Revolute joint between 1 and 5 fails separated. 5 bar chain	

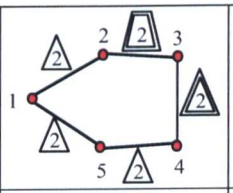
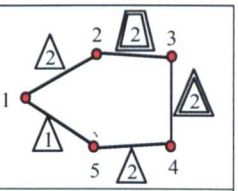
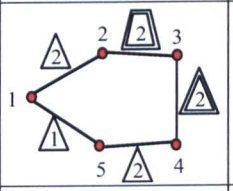
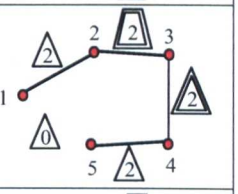
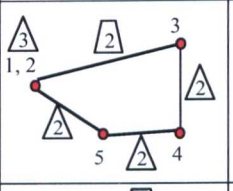
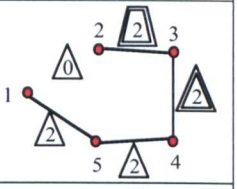
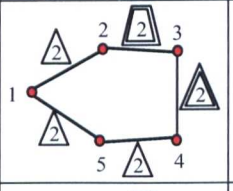
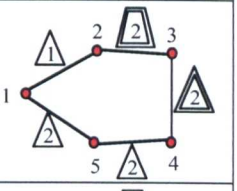
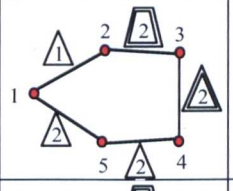
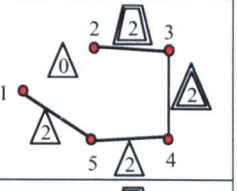
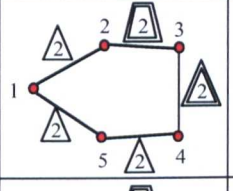
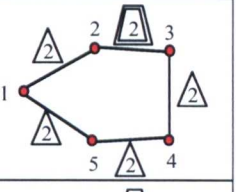
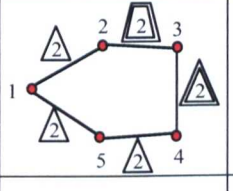
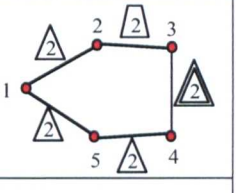
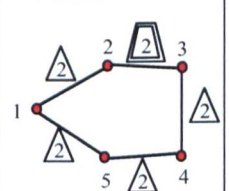
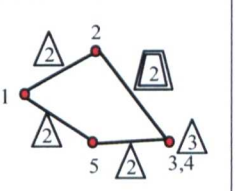
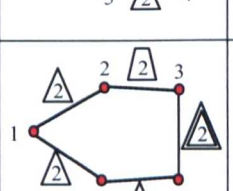
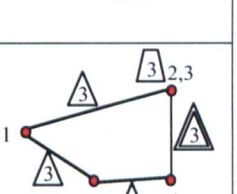
S1	Nominal – fully functional		S13	Revolute joint between 1 and 5 gains 1 dof. 5 loop 'without loss of edge'	
S13	Revolute joint between 1 and 5 gains 1 dof		S12	Revolute joint between 1 and 5 fails separated. 5 bar chain	
S6	Revolute joint between 1 and 2 locks up, 1 and 2 become coincident		S14	Revolute joint between 1 and 2 fails separated. 5 bar chain	
S1	Nominal – fully functional		S15	Revolute joint between 1 and 2 gains 1 dof. 5 loop 'without loss of edge'	
S15	Revolute joint between 1 and 2 gains 1 dof		S14	Revolute joint between 1 and 2 fails separated. 5 loop 'without loss of edge'	
S1	Nominal – fully functional		S16	Drive fails in revolute joint 3 / 4. System acts as four bar linkage and collapses. 5 loop 'without loss of edge'	
S1	Nominal – fully functional		S18	Drive fails in prismatic joint between 2 and 3. 5 loop 'without loss of edge'	
S16	Drive fails in revolute joint between 3 and 4. System acts as four bar linkage and collapses		S17	Lock revolute joint between 3 and 4 to regain limited functionality. 5 loop 'without loss of edge'	
S18	Drive fails in prismatic joint between 2 and 3.		S19	Lock prismatic joint between 2 and 3 to regain limited functionality. 5 loop 'without loss of edge'	

Table 8.10: State Table for System Based on Attila Leg Mechanism

From the above, the corresponding fault state digraph can be derived:

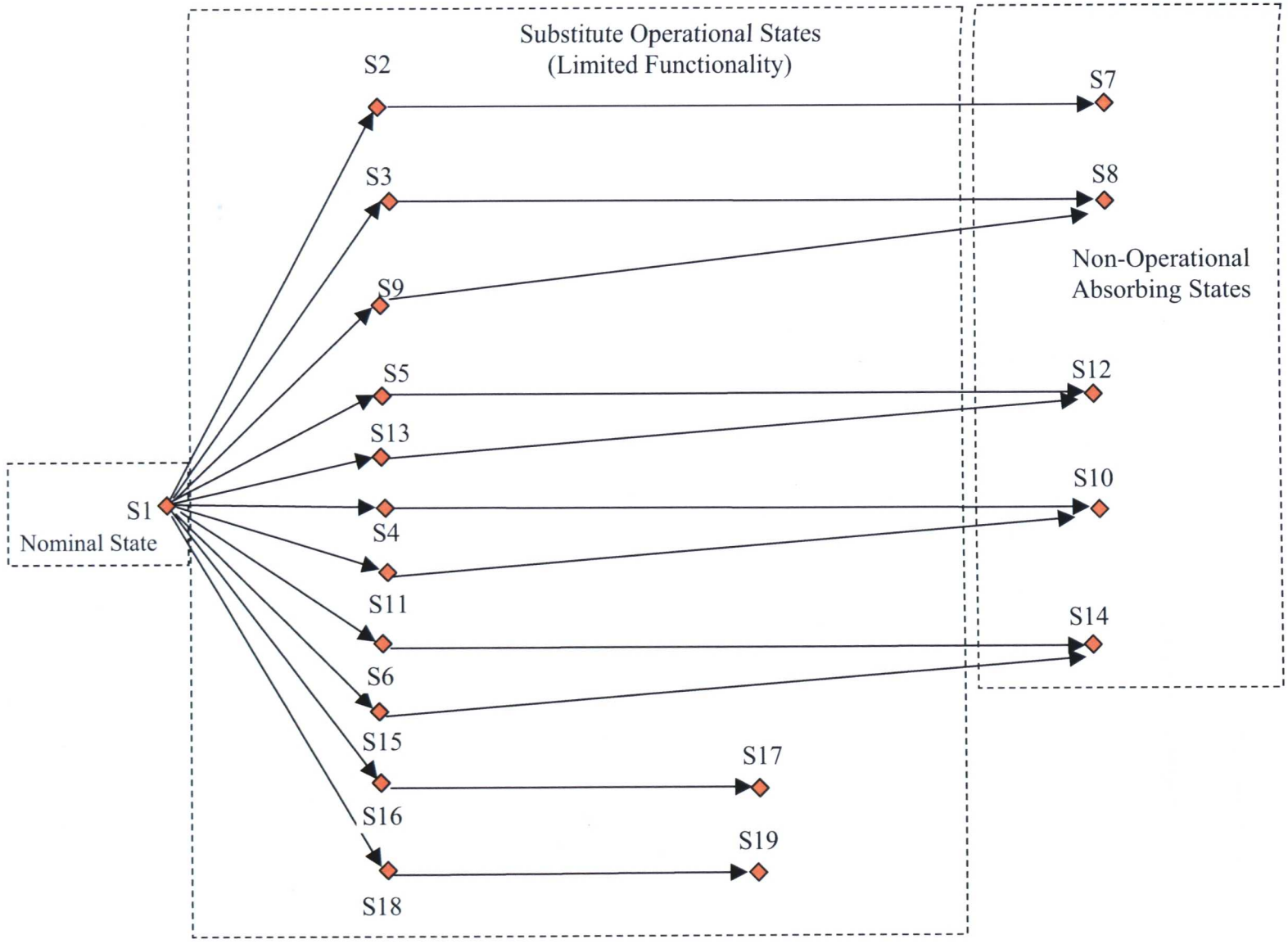


Figure 8.10:Fault State Digraph for System Based on Attila Leg Mechanism

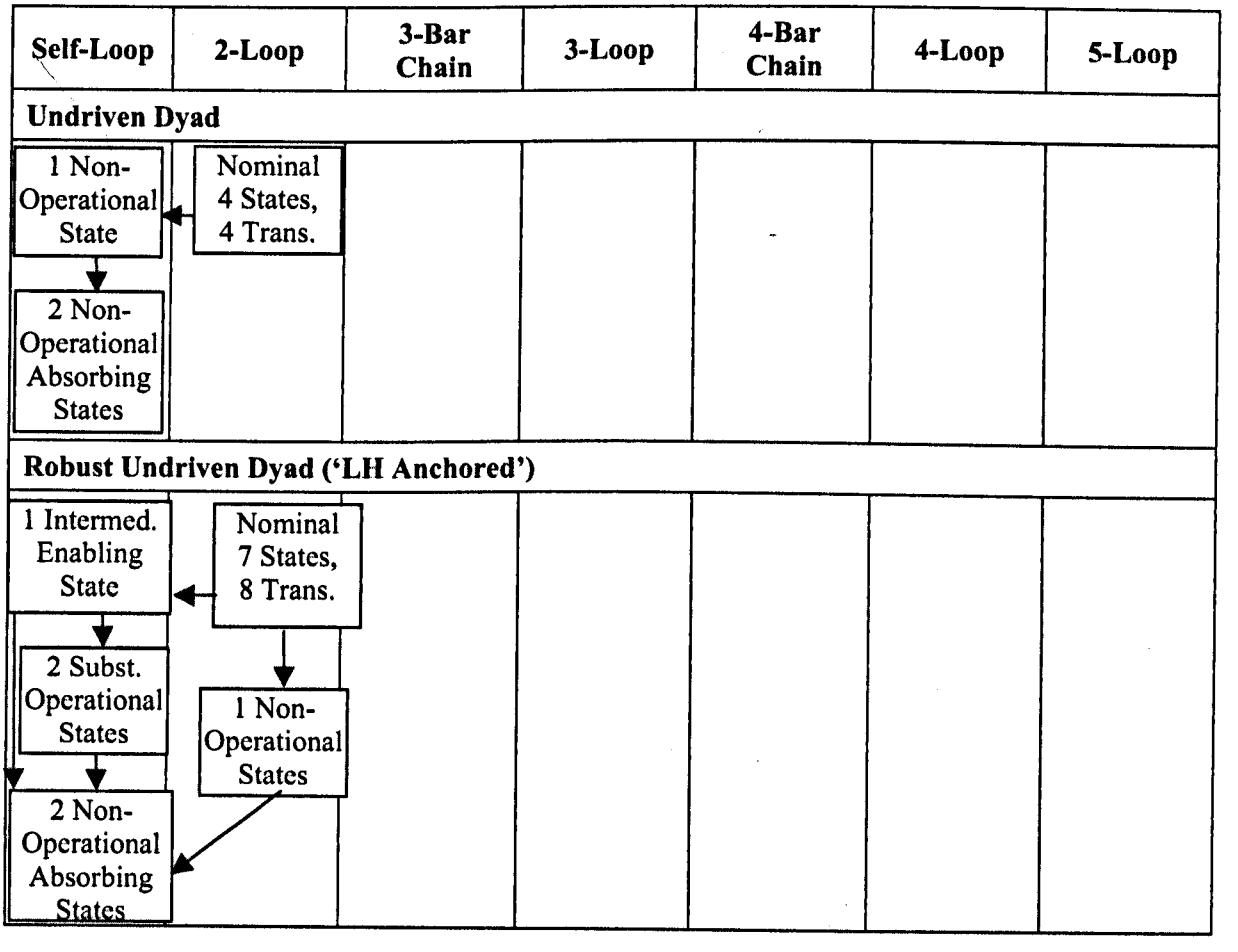
8.3.2 Classes of System Fault State

On considering the fault state digraphs presented in the previous section, it can be seen that when the different fault configurations of any one system are considered collectively, they form groupings according to the fault path that they lie on, and also according to the category, as defined in 8.3.1.1, that they fall into. – that is *substitute operational state*, *non-operational state* etc..

This grouping is referred to here as their **System Fault State Category.** The Fault State Categories are as defined earlier, viz:

- Nominal State
- Substitute Operational State
- Intermediate Enabling State
- Non-Operational State
- Non-Operational Absorbing State

The appropriate categories applicable to any one system are identified (by 'boxing' – shown dashed) in the relevant foregoing digraphs. By extracting these groupings from the digraphs, it is possible to derive a table of system fault state categories, as shown in Table 8.11.



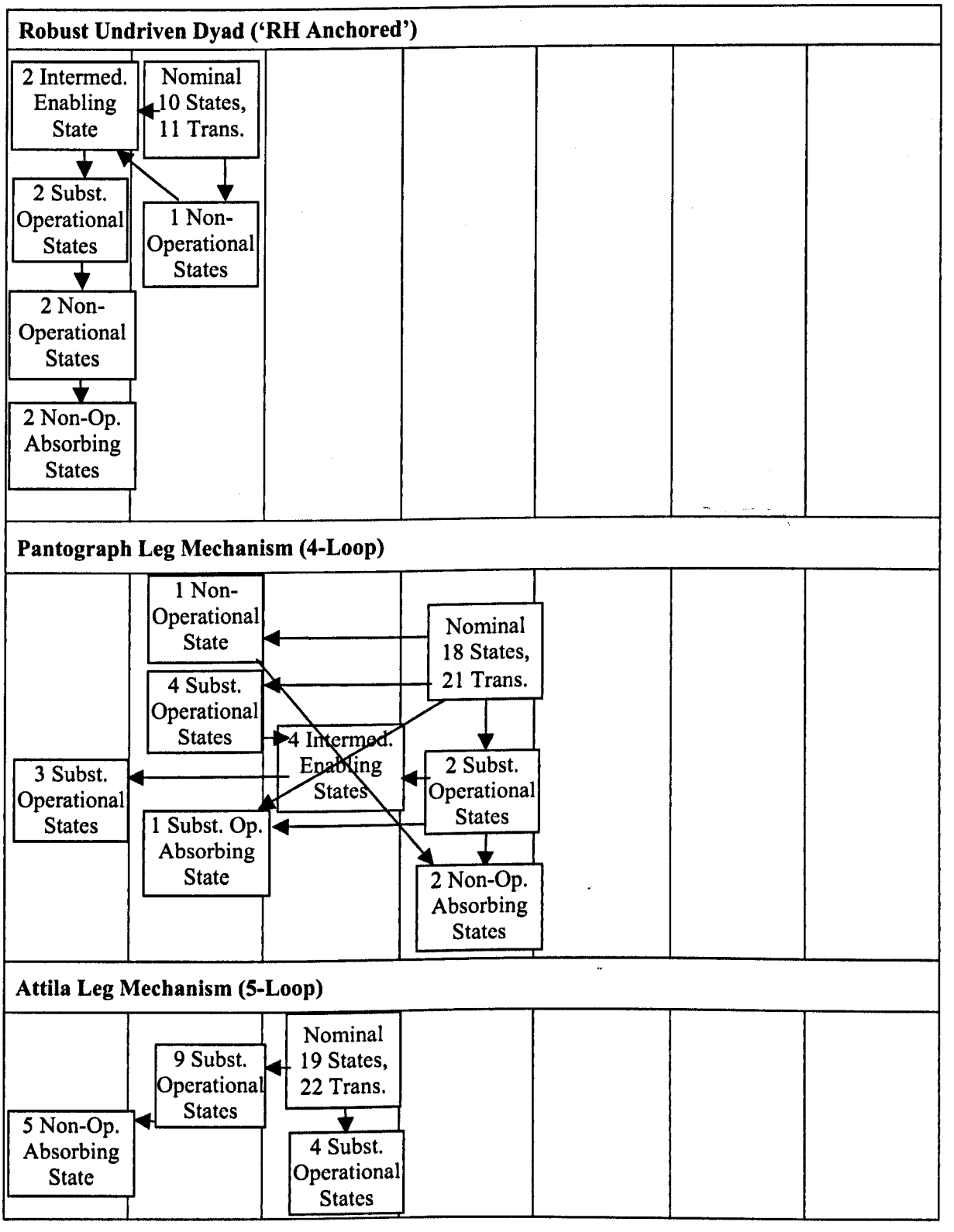
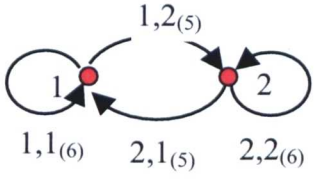
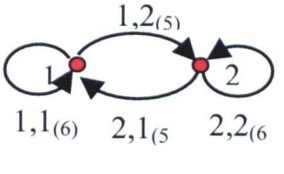
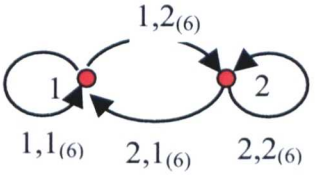
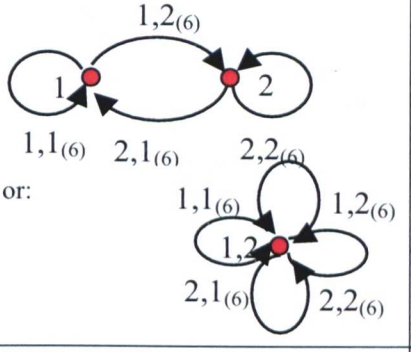
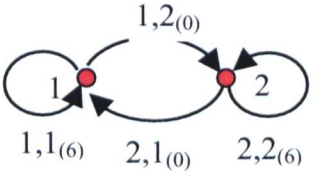
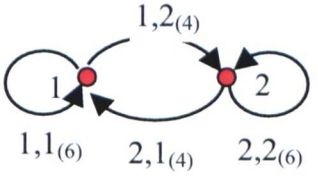
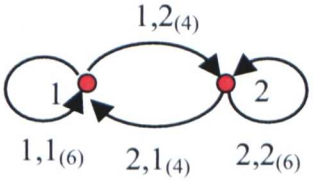


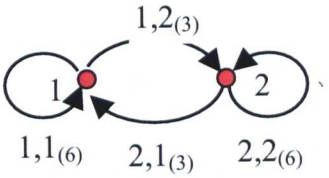
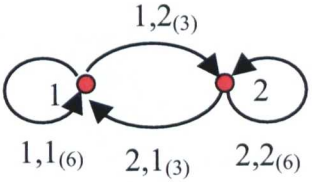
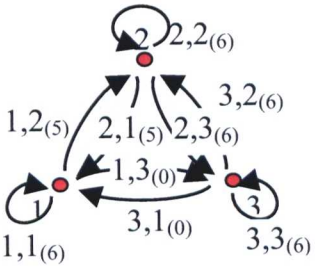
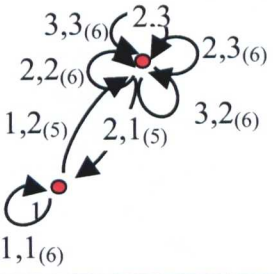
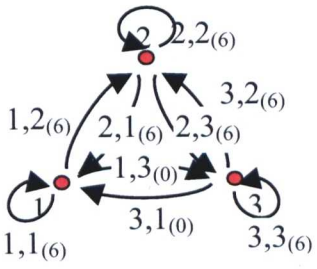
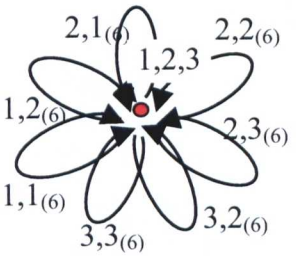
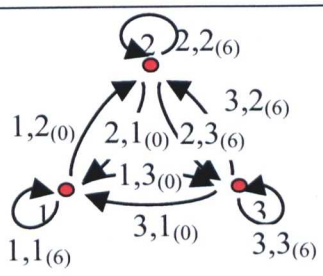
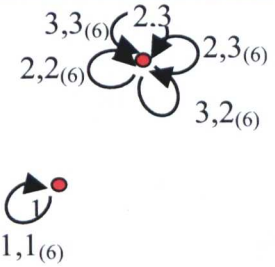
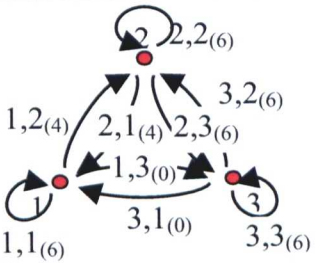
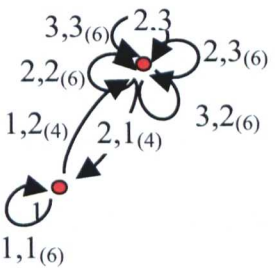
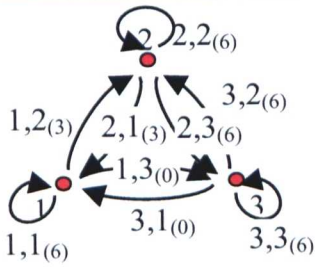
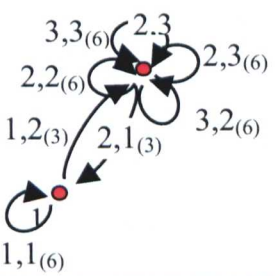
Table 8.11: System Fault Class Categories for Example 2, 3 and 4-Bar Systems

It will be seen that any one system fault state will belong to a Fault State Category because of its potential, or lack of potential, for further change. Such system states can have a number of structures, even though they belong in the same category. These structures recur because there are only a finite number of permutations of characteristic matrix terms. These recurring structures are

referred to as **Fault Classes**. They are related to the root node (nominal) system by the relationship between the root node characteristic polynomial, and that of the fault state. Thus, one can regard a fault class as being derived from the nominal system configuration by removal (addition?) of specific symbolic determinant terms, and that traversing the system fault state digraph is by this means.

This can be illustrated by considering the various states identified earlier for the 'Undriven Dyad' and 'Robust Undriven Dyad (LH Anchored)' systems, and presenting them in a format based on the loop relationships within the systems, using the notation defined in 8.1.1. The results are presented as Table 8.12.

STATE No.	COMPLETE SYSTEM REPRESENTATION	REDUCED VERSION	CONSTRAINTS MATRIX / DETERMINANT
Undriven Dyad			
S1			$\begin{bmatrix} 6 & 5 \\ 5 & 6 \end{bmatrix}$ $(d_{11})(d_{22}) - (d_{12}\, d_{21})$
S2			$\begin{bmatrix} 6 & 6 \\ 6 & 6 \end{bmatrix}$ $(d_{11})(d_{22}) - (d_{12}\, d_{21})$
S3			$\begin{bmatrix} 6 & 0 \\ 0 & 6 \end{bmatrix}$ $(d_{11})(d_{22})$
S4			$\begin{bmatrix} 6 & 4 \\ 4 & 6 \end{bmatrix}$ $(d_{11})(d_{12}) - (d_{12}\, d_{21})$

S5			$\begin{bmatrix} 6 & 3 \\ 3 & 6 \end{bmatrix}$ $(d_{11})(d_{12}) - (d_{12}\, d_{21})$
Robust Dyad ('LH Anchored')			
S1			$\begin{bmatrix} 6 & 5 & 0 \\ 5 & 6 & 6 \\ 0 & 6 & 6 \end{bmatrix}$ $(d_{11})(d_{22})(d_{33}) - (d_{11})(d_{23}\, d_{32}) - (d_{33})(d_{12}\, d_{21})$
S2			$\begin{bmatrix} 6 & 6 & 0 \\ 6 & 6 & 6 \\ 0 & 6 & 6 \end{bmatrix}$ $(d_{11})(d_{22})(d_{33}) - (d_{11})(d_{23}d_{32}) - (d_{33})(d_{12}\, d_{21})$
S3			$\begin{bmatrix} 6 & 0 & 0 \\ 0 & 6 & 6 \\ 0 & 6 & 6 \end{bmatrix}$ $(d_{11})(d_{22})(d_{33}) - (d_{11})(d_{23}\, d_{32})$
S4			$\begin{bmatrix} 6 & 4 & 0 \\ 4 & 6 & 6 \\ 0 & 6 & 6 \end{bmatrix}$ $(d_{11})(d_{22})(d_{33}) - (d_{11})(d_{23}\, d_{32}) - (d_{33})(d_{12}\, d_{21})$
S5			$\begin{bmatrix} 6 & 3 & 0 \\ 3 & 6 & 6 \\ 0 & 6 & 6 \end{bmatrix}$ $(d_{11})(d_{22})(d_{33}) - (d_{11})(d_{23}\, d_{32}) - (d_{33})(d_{12}\, d_{21})$

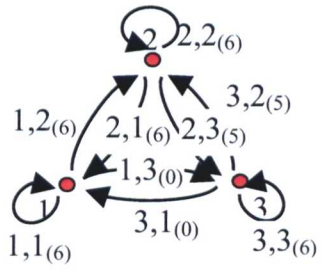
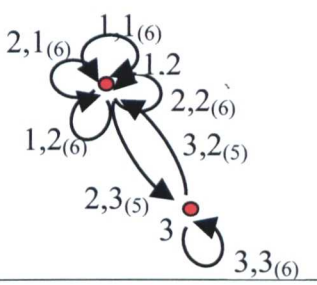
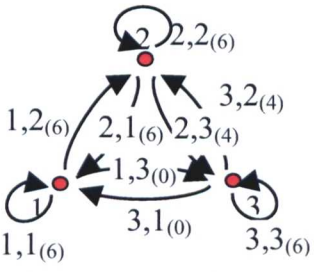
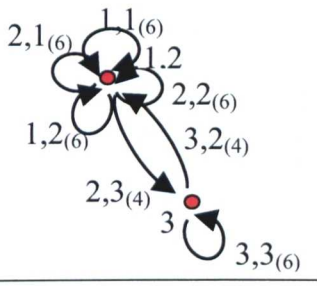
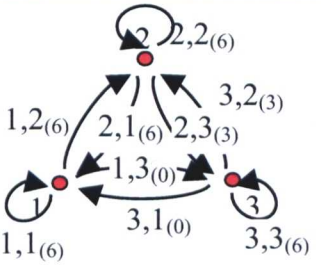
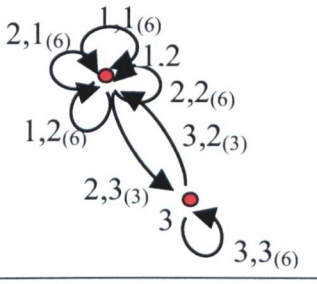
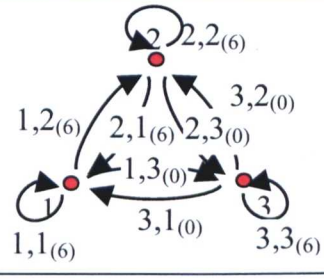
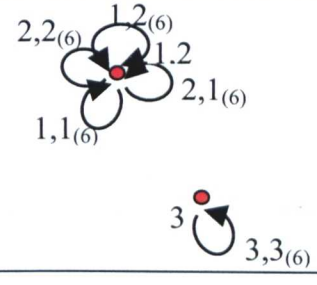
S6			$\begin{bmatrix} 6 & 6 & 0 \\ 6 & 6 & 5 \\ 0 & 5 & 6 \end{bmatrix}$ $(d_{11})(d_{22})(d_{33}) - (d_{11})(d_{23}\,d_{32}) - (d_{33})(d_{12}\,d_{21})$
S7			$\begin{bmatrix} 6 & 6 & 0 \\ 6 & 6 & 4 \\ 0 & 4 & 6 \end{bmatrix}$ $(d_{11})(d_{22})(d_{33}) - (d_{11})(d_{23}\,d_{32}) - (d_{33})(d_{12}\,d_{21})$
S8			$\begin{bmatrix} 6 & 6 & 0 \\ 6 & 6 & 3 \\ 0 & 5 & 6 \end{bmatrix}$ $(d_{11})(d_{22})(d_{33}) - (d_{11})(d_{23}\,d_{32}) - (d_{33})(d_{12}\,d_{21})$
S9			$\begin{bmatrix} 6 & 6 & 0 \\ 6 & 6 & 0 \\ 0 & 0 & 6 \end{bmatrix}$ $(d_{11})(d_{22})(d_{33}) - (d_{33})(d_{12}\,d_{21})$

Table 8.12: 'Loop' representation of System States

8.4 Interpretation of the Classes of System Fault State

8.4.1 Symbolic Interpretation of Characteristic Polynomials

The previous parts of this chapter showed that when systems degrade, the resulting configurations will fall into one fault class or another. It is therefore useful to investigate more closely the way in which this degradation occurs, using the additional representations that have been created, in order to judge whether the methods and representations established in Chapter 5 and Chapter 7 can be further developed, and whether an integrated view of system behaviour under fault action can be established.

Consider the form of the various terms encountered in the characteristic polynomials of two-, three- and four-bar systems. Applying the symbology defined in 8.1.1, it is possible to ascribe additional

meaning to the symbolic terms of the determinants of the characteristic polynomials of these systems, as shown in Table 8.13.

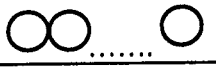
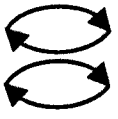
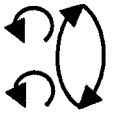
Form of CP Term	Symbol	Meaning
$(d_{11} + d_{22} + ...+ d_{nn})$		This is the number of nodes in the system, for each of which there is a self-loop.
$(d_{11})(d_{22})...(d_{nn})$		These are the combinations in which the nodes in the system are connected.
$(d_{12}d_{21})$		These are the different 2-loops which exist in the system.
$(d_{11})(d_{23}d_{32})$		This is a relationship that can exist in a system with three or more nodes – one node acts only as a self-loop, the other two nodes act as a 2-loop, which is not connected to the self-loop node.
$(d_{12}d_{23}d_{31})$		This is a relationship that can exist in a system with three or more nodes – three nodes together operate as a single, directed, 3-loop.
$(d_{12}d_{21})(d_{34}d_{43})$		This is a relationship that can exist in a system with four or more nodes – two pairs of nodes operate as two separated 2-loops.
$(d_{11})(d_{22})(d_{34}d_{43})$		This is a relationship that can exist in a system with four or more nodes – four nodes operate as one separated 2-loop plus two separated self-loops.
$(d_{11})(d_{32}d_{24}d_{43})$		This is a relationship that can exist in a system with four or more nodes – four nodes operate as one directed 3-loop plus one separated self-loop.
$(d_{14}d_{43}d_{32}d_{21})$		This is a relationship that can exist in a system with four or more nodes – four nodes operate as a single directed four-loop.
$(d_{21}d_{13}d_{34}d_{42})$		This is a relationship that can exist in a system with four or more nodes – four nodes operate as a single *twisted* directed four-loop.

Table 8.13: Meaning of Characteristic Polynomial Terms

Having established a working hypothesis for determinant term meaning, the defined symbology can be applied to the characteristic polynomials of two-, three- and four-bar systems.

The method of application of the symbology is based on the pictorial interpretation of any given characteristic polynomial by substitution of the appropriate relationship symbols for the polynomial terms. This allows an 'at a glance' appreciation of the polynomial structure to be formed, and, as shown later, is amenable to manipulation in order to represent the evolution of faults states in terms of the loss of relationships.

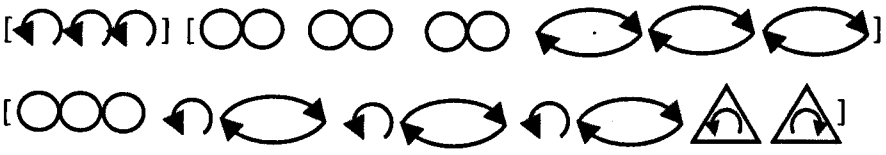
Characteristic Polynomial of a complete two-bar system:

$\lambda^2 - (d_{11} + d_{22})\lambda + \{(d_{11})(d_{22}) - (d_{12}\, d_{21})\}$

There are two nodes. A relationship exists between these two nodes. The relationship takes the form of a 2-loop.

Characteristic Polynomial of a complete three-bar system:

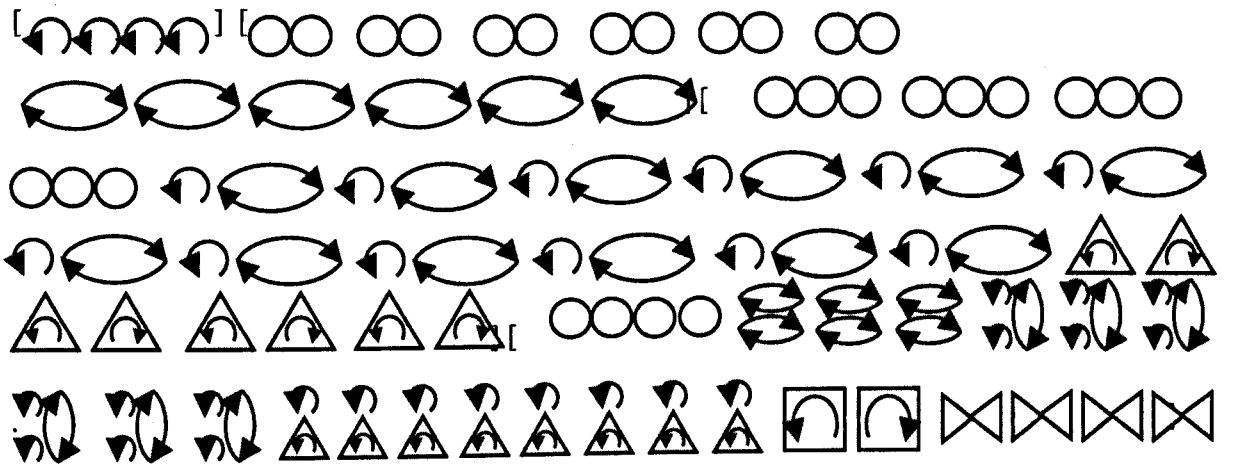
$\lambda^3 - (d_{11} + d_{22} + d_{33})\lambda^2 + \{(d_{11})(d_{33}) + (d_{22})(d_{33}) + (d_{11})(d_{22}) - (d_{13}\, d_{31}) - (d_{12}\, d_{21}) - (d_{23}\, d_{32})\}\, \lambda + \{-(d_{11}\, d_{22}\, d_{33}) + (d_{22})(d_{13}\, d_{31}) + (d_{11})(d_{23}\, d_{32}) + (d_{33})(d_{12}\, d_{21}) - (d_{12}\, d_{23}\, d_{31}) - (d_{12}\, d_{23}\, d_{31})\}$



There are three nodes. Three relationships exist between all possible pairs of these three nodes. The relationship between the pairs takes the form of three 2-loops. Relationships also exist between the three nodes of the system when acting together. These take the form of three 'self-loop plus separated 2-loop' relationships, plus two opposite signed 3-loops.

Characteristic Polynomial of a complete four-bar system:

$\lambda^4 - (d_{11} + d_{22} + d_{33} + d_{44})\lambda^3 + \{(d_{11})(d_{33}) + (d_{11})(d_{44}) + (d_{33})(d_{44}) + (d_{22})(d_{44}) + (d_{22})(d_{33}) + (d_{11})(d_{22}) - (d_{34}\, d_{43}) - (d_{32}\, d_{23}) - (d_{42}\, d_{24}) - (d_{21}\, d_{12}) - (d_{31}\, d_{13}) - (d_{41}\, d_{14})\}\lambda^2 + \{-(d_{22})(d_{33})(d_{44}) - (d_{11})(d_{33})(d_{44}) - (d_{11})(d_{22})(d_{44}) - (d_{11})(d_{22})(d_{33}) + (d_{11})(d_{34}\, d_{43}) + (d_{22})(d_{34}\, d_{43}) + (d_{44})(d_{32}\, d_{23}) + (d_{32}\, d_{24}\, d_{43}) - (d_{42}\, d_{23}\, d_{34}) + (d_{33})(d_{42}\, d_{24}) + (d_{11})(d_{32}\, d_{23}) + (d_{44})(d_{21}\, d_{12}) + (d_{33})(d_{21}\, d_{12}) + (d_{21}\, d_{32}\, d_{13}) + (d_{21}\, d_{42}\, d_{14}) + (d_{31}\, d_{12}\, d_{23}) + (d_{44})(d_{31}\, d_{13}) + (d_{31}\, d_{14}\, d_{43}) + (d_{22})(d_{31}\, d_{13}) + (d_{41}\, d_{12}\, d_{24}) + (d_{41}\, d_{13}\, d_{34}) + (d_{33})(d_{41}\, d_{14}) + (d_{22})(d_{41}\, d_{14})\}\lambda + \{(d_{44})(d_{22})(d_{33})(d_{44}) + (d_{21}\, d_{12})(d_{34}\, d_{43}) + (d_{31}\, d_{13})(d_{42}\, d_{24}) + (d_{41}\, d_{14})(d_{32}\, d_{23}) + (d_{11})(d_{22})(d_{34}\, d_{43}) + (d_{11})(d_{44})(d_{32}\, d_{23}) + (d_{11})(d_{32}\, d_{24}\, d_{43}) - (d_{11})(d_{42}\, d_{23}\, d_{34}) - (d_{11})(d_{33})(d_{42}\, d_{24}) - (d_{33})(d_{44})(d_{21}\, d_{12}) - (d_{44})(d_{21}\, d_{32}\, d_{13}) - (d_{21}\, d_{32}\, d_{14}\, d_{43}) - (d_{21}\, d_{42}\, d_{13}\, d_{34}) - (d_{33})(d_{21}\, d_{42}\, d_{14}) - (d_{44})(d_{31}\, d_{12}\, d_{23}) - (d_{31}\, d_{12}\, d_{24}\, d_{43}) - (d_{22})(d_{44})(d_{31}\, d_{13}) - (d_{22})(d_{31}\, d_{14}\, d_{43}) - (d_{31}\, d_{42}\, d_{14}\, d_{23}) - (d_{41}\, d_{12}\, d_{23}\, d_{34}) - (d_{33})(d_{41}\, d_{12}\, d_{24}) - (d_{22})(d_{41}\, d_{13}\, d_{34}) - (d_{22})(d_{33})(d_{41}\, d_{14}) - (d_{41}\, d_{32}\, d_{13}\, d_{24})\}$



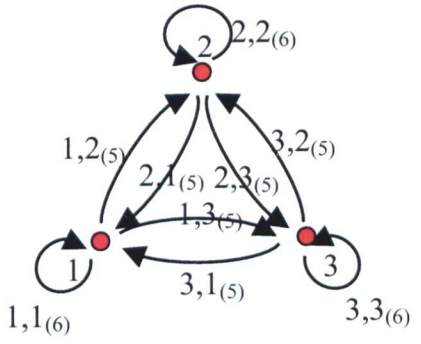
There are four nodes. Six relationships exist between all possible pairs of these four nodes. The relationship between the pairs takes the form of six 2-loops. Relationships also exist between the four permutations of three nodes of the system. These take the form of twelve 'self-loop plus separated 2-loop' relationships, plus eight opposite signed 3-loops. Further functional relationships exist between all four nodes acting together. These are three 'two separated 2-loops', six 'two self loops plus separated 2-loop, eight 'directed 3-loop plus separated self-loop', two oppositely directed 4-loops, and four oppositely directed 'twisted 4-loops'.

8.4.2 Method Applied to Fault States

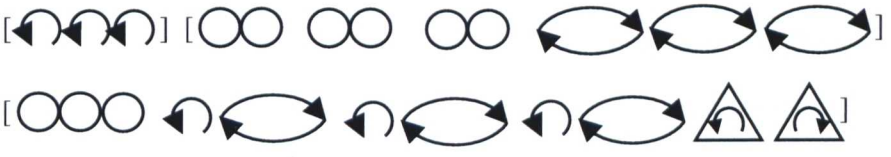
When applying the method to an example system that is not, in itself, a complete graph, the system must be considered in three steps, as follows:

- Identify all determinant terms – the terms that would be present in the complete graph should be identified.
- Identify nominal determinant terms – the terms present in the nominal (ie unfailed) system should be identified, so that it is possible to state:
  - terms missing compared with complete graph
  - starting point (ie nominal state) terms
- Identify fault state terms – the fault state being considered can be defined as a combination of:
  - Lost terms
  - Remaining terms

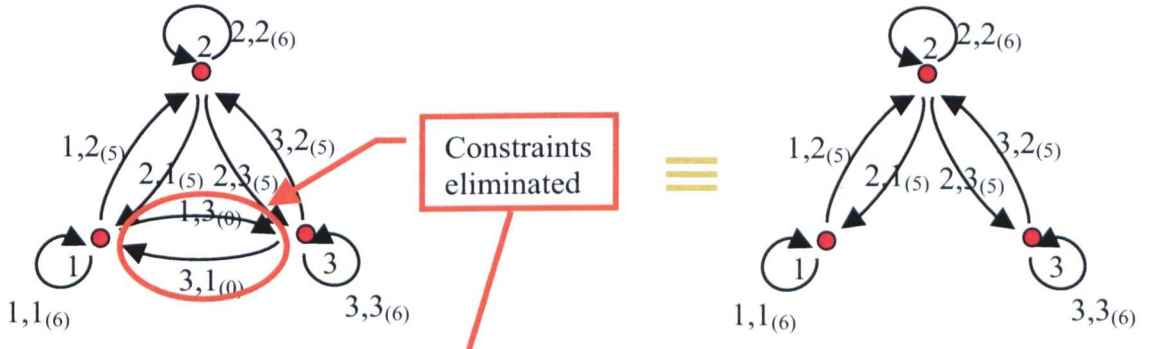
This methodology is now applied to the example case of the 'Robust Dyad ('LH Anchored') discussed previously, by first identifying the Complete Graph relationships:



The complete graph for the system, assuming 1 dof joints, is shown above. The relationships for a three-bar complete system were identified in the previous section as follows:

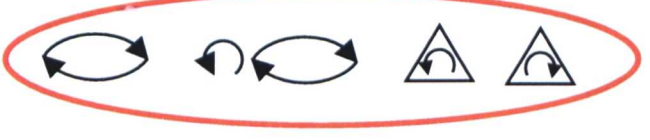


By eliminating any relationships involving the 2-loop ($d_{13}d_{31}$), the graphical representation shown below is obtained:



Thus, the relationships for the nominal condition of the 'Robust Dyad ('LH Anchored') can be shown to be:

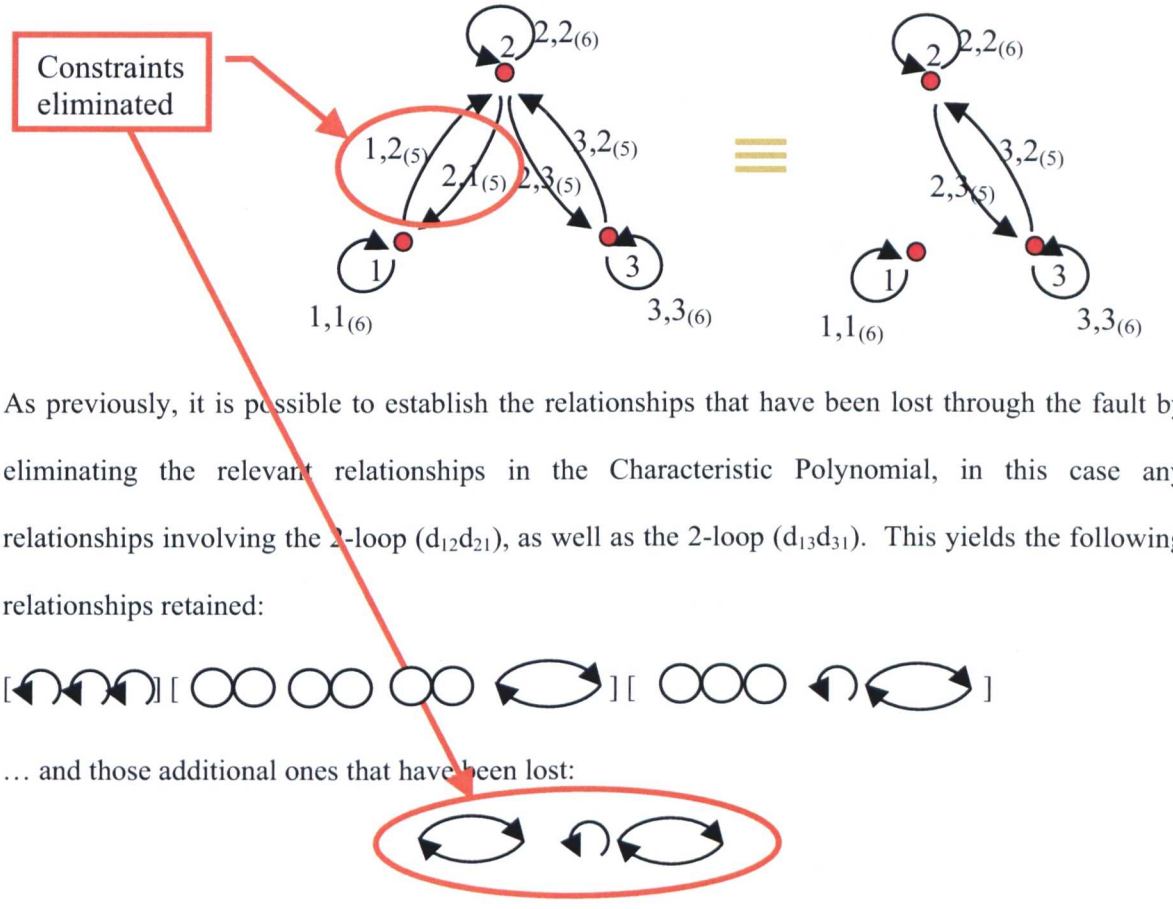
Which means that the relationships missing compared with the complete graph are as follows:



This is logical, since it shows that relationships that depend on the presence of a complete 3-Loop, and those that depend on the presence of 2-loop ($d_{13}d_{31}$) are eliminated.

It is now possible to proceed to apply this method to a fault state. The case chosen represents the loss of all constraint between vertices 1 and 2, that is to say that the graph becomes disconnected.

In fact, using the methodology applied earlier, it is not possible to move immediately to this state without passing through intermediate states representing progressive loss of constraint. The state is chosen for clarity, and the nature of the path used to arrive at it is not central to this discussion.



As previously, it is possible to establish the relationships that have been lost through the fault by eliminating the relevant relationships in the Characteristic Polynomial, in this case any relationships involving the 2-loop ($d_{12}d_{21}$), as well as the 2-loop ($d_{13}d_{31}$). This yields the following relationships retained:

… and those additional ones that have been lost:

8.4.3 Commentary

This chapter has derived a method whereby any fault state can be expressed in terms of lost relationships that are determined by the structure of the characteristic polynomial. This method makes it possible to group potential system fault states into fault classes, based on the retention or loss of specific relationships. The fault classes into which a system can degenerate will be used later in Chapter 10 as a significant discriminator in the comparison of systems.

The results of this chapter can also be correlated with those of Chapter 5, where the behaviour of eigenvectors during transition along a fault path under the influence of sequential faults was investigated. Tables 5.3 and 5.4 in Chapter 5 showed that there is only a limited set of symbolic determinants that controls the behaviour of the eigenvectors. These are depicted in Table 8.14

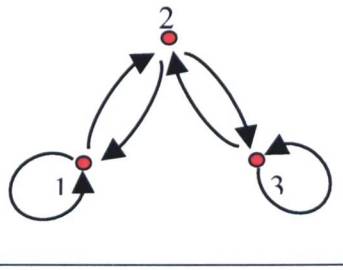
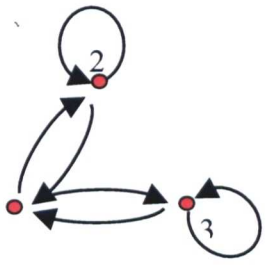
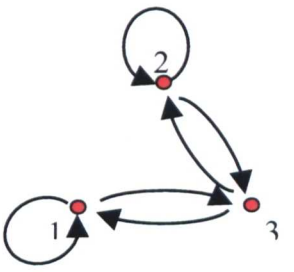
		
		
$(d_{11})(d_{22})(d_{33}) - (d_{11})(d_{23}d_{32}) - (d_{33})(d_{12}d_{21})$	$(d_{11})(d_{22})(d_{33}) - (d_{22})(d_{13}d_{31}) - (d_{33})(d_{12}d_{21})$	$(d_{11})(d_{22})(d_{33}) - (d_{11})(d_{23}d_{32}) - (d_{22})(d_{13}d_{31})$

Table 8.14: Diagrammatic Representation of Example 3-Bar System Fault Classes

The table illustrates the fact derived in Chapter 5 that the fault class is unchanged if only the degrees of freedom change. Only when there is 'loss of edge' or 'vertex identification' is there a change in fault class. This can be seen by examining the fault paths discussed in Chapter 5, where Tables 5.3 and 5.4 show that there is no change in the symbolic determinants of the systems as they transition along the fault path. The expanded determinants referred to in these two tables are those used as examples in Table 8.14.

Chapter 9
Distinguishing Co-Spectral Systems

Chapter 8 developed a methodology for representing the degeneration of systems under the action of faults, using the terms resulting from symbolic evaluation of the relevant characteristic polynomials. In order to be able to apply this representation with confidence, an ability to deal with co-spectral graphs is important, since any system comparison is uncertain without a means of discriminating between different systems having the same spectrum. This chapter, therefore, proceeds to examine the proposed methodology in the context of this particular category of system.

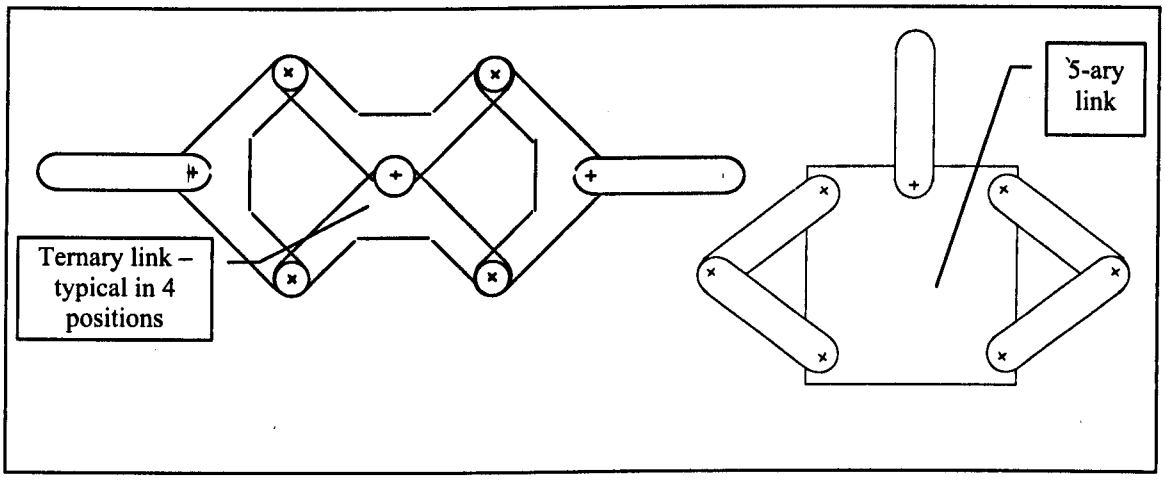
9.1 The Characteristics of Co-Spectral Graphs

Consider the two planar systems, illustrated in Figure 9.1, below. It is easily demonstrated that the characteristic polynomials for both graphs, when derived from their Adjacency Matrices, **A**, are identical (see Table 9.1). Moreover, the characteristic polynomials derived from their Constraints Matrices, **C**, are also identical. Thus System (a) and System (b) are cospectral.

The systems that these interchange graphs represent are, however, quite different, as can be seen by comparing their direct graphs and mobilities. Since the systems involve ternary and 5-ary links, it is necessary to derive the system mobilities using the 'Second Mobility Criterion' [54] that defines mobility as follows:

$$M = 2g + 3n_o + n_1 - ((2r-3)n_r + \ldots\ldots + (2s-3)n_s) - 3 \quad \ldots\ldots\ldots\ldots (9.1)$$

where g is the total number of joints (regardless of their multiplicity), no link has multiplicity greater than s, n_o is the number of nullary links, n_1 is the number of unary links, and n_r is the number of r-ary links ($2 \leq r \leq s$).



System (a)
(2 unary links, 4 ternary links)

System (b)
(1 unary link, 4 binary links, 1 5-ary link)

Figure 9.1: Schematic of two Co-Spectral Systems

Applying Expression 9.1, it can be shown that System (a) has a mobility of 2, whilst System (b) has a mobility of 1. (Note that application of expression 9.1 does, in fact, give a mobility of 1 for system (a). The reason for this is that the system is over-constrained. By analysis of the system as an over-constrained, central subsystem in isolation from the two unary links, it can be shown that this central subsystem has a mobility of –1, ie immobile, so that the true mobility of the overall system is 2 when the unary links are added back in. This is a common problem in systems where account needs to be taken of over- or under-constraint. More sophisticated mobility expressions are able to allow for this).

Whilst the mobility result obtained is, possibly, the intuitive answer, the result shows that although the two systems have cospectral interchange graphs, the behaviour of the systems is different.

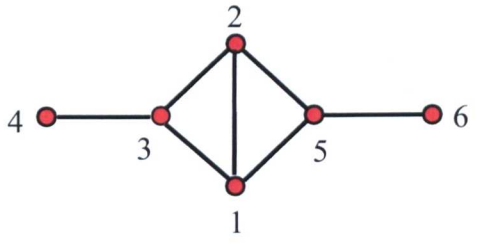
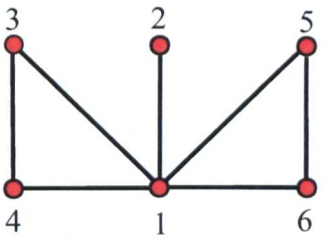
	
Characteristic polynomial based on **A** is $\lambda^6 - 7\lambda^4 - 4\lambda^3 + 7\lambda^2 + 4\lambda - 1$	
Characteristic polynomial based on **C** is $\lambda^6 - 36\lambda^5 + 365\lambda^4 - 620\lambda^3 - 4985\lambda^2 + 10544\lambda - 5269$	
Number of terms in Characteristic Polynomial of complete graph = 6! = 720	
Number of terms in Characteristic Polynomial based on **A** = 27	Number of terms in Characteristic Polynomial based on **A** = 23
Number of terms in Characteristic Polynomial based on **C** = 260	Number of terms in Characteristic Polynomial based on **C** = 244
System (a)	*System (b)*

Table 9.1: Some Representative Data for the two Co-Spectral Graphs Considered

The fact that these systems are only superficially similar, and are, in fact, quite different, is shown in Tables 9.2 and 9.3. These tables show, that the identical characteristic polynomials are derived from non-identical subsets of terms derived from those for the characteristic polynomial of the complete graph on six vertices.

Note: In the context of this work, 'term' is taken to mean a term in the expanded determinant of **A** or **C** before grouping into characteristic polynomial coefficients.

For definition of the symbology used in Tables 9.2 and 9.3, refer to Section 8.1.1.

9.2 Terms derived from the Adjacency Matrix CP

Investigation shows that, before grouping terms into coefficients, the Characteristic Polynomial based on **A** for System (a) has 27 terms, whilst that for System (b) has 23 terms, as shown in Table 9.2. There are two more paired 2-loops in (a) compared with (b), and (a) also has two 4-loops which are not present in (b). Note: the differences recorded are in the TOTAL numbers of terms of any one type.

SYSTEM (a)		SYSTEM (b)	
Term	**Icon**	**Term**	**Icon**
λ^6		λ^6	
λ^4		λ^4	
$-(d_{23}\, d_{32})$			
$-(d_{25}\, d_{52})$			
$-(d_{12}\, d_{21})$		$-(d_{12}\, d_{21})$	
$-(d_{13}\, d_{31})$		$-(d_{13}\, d_{31})$	
$-(d_{56}\, d_{65})$		$-(d_{56}\, d_{65})$	
$-(d_{34}\, d_{43})$		$-(d_{34}\, d_{43})$	
$-(d_{15}\, d_{51})$		$-(d_{15}\, d_{51})$	
		$-(d_{14}\, d_{41})$	
		$-(d_{16}\, d_{61})$	

λ^3		λ^3	
$-(d_{12}\, d_{25}\, d_{51})$			
$-(d_{21}\, d_{52}\, d_{15})$			
$-(d_{12}\, d_{23}\, d_{31})$			
$-(d_{21}\, d_{32}\, d_{13})$			
		$-(d_{13}\, d_{34}\, d_{41})$	
		$-(d_{31}\, d_{43}\, d_{14})$	
		$-(d_{15}\, d_{56}\, d_{61})$	
		$-(d_{51}\, d_{65}\, d_{16})$	

λ^2		λ^2	
$(d_{23}\ d_{32})(d_{56}\ d_{65})$			
$(d_{15}\ d_{51})(d_{23}\ d_{32})$			
$(d_{13}\ d_{31})(d_{25}\ d_{52})$			
$(d_{34}\ d_{43})(d_{25}\ d_{52})$			
$(d_{34}\ d_{43})(d_{15}\ d_{51})$		$(d_{34}\ d_{43})(d_{15}\ d_{51})$	
$(d_{34}\ d_{43})(d_{56}\ d_{65})$		$(d_{34}\ d_{43})(d_{56}\ d_{65})$	
$(d_{12}\ d_{21})(d_{56}\ d_{65})$		$(d_{12}\ d_{21})(d_{56}\ d_{65})$	
$(d_{12}\ d_{21})(d_{34}\ d_{43})$		$(d_{12}\ d_{21})(d_{34}\ d_{43})$	
$(d_{13}\ d_{31})(d_{56}\ d_{65})$		$(d_{13}\ d_{31})(d_{56}\ d_{65})$	
		$(d_{14}\ d_{41})(d_{56}\ d_{65})$	
		$(d_{16}\ d_{61})(d_{34}\ d_{43})$	
$-(d_{23}\ d_{31}\ d_{15}\ d_{52})$			
$-(d_{32}\ d_{13}\ d_{51}\ d_{25})$			

λ		λ	
$(d_{34}\ d_{43})(d_{12}\ d_{25}\ d_{51})$			
$(d_{34}\ d_{43})(d_{21}\ d_{52}\ d_{15})$			
$(d_{56}\ d_{65})(d_{12}\ d_{23}\ d_{31})$			
$(d_{56}\ d_{65})(d_{21}\ d_{32}\ d_{13})$			
		$(d_{34}\ d_{43})(d_{15}\ d_{56}\ d_{61})$	
		$(d_{34}\ d_{43})(d_{51}\ d_{65}\ d_{16})$	
		$(d_{56}\ d_{65})(d_{13}\ d_{34}\ d_{41})$	
		$(d_{56}\ d_{65})(d_{31}\ d_{43}\ d_{14})$	

Constant terms		**Constant terms**	
$(d_{12}\ d_{21})(d_{34}\ d_{43})(d_{56}\ d_{65})$		$(d_{12}\ d_{21})(d_{34}\ d_{43})(d_{56}\ d_{65})$	

Table 9.2:Comparison of CP Terms of the two Co-Spectral Graphs Considered, based on **A**

9.3 Terms Derived from the Constraints Matrix CP

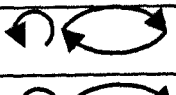
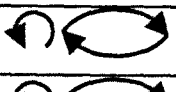
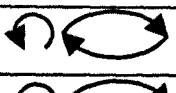
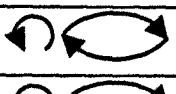
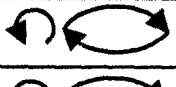
The Characteristic Polynomial based on the Constraints Matrix, **C**, for System (a) has 260 terms before grouping terms into coefficients, whilst that for System (b) has 244 terms, as shown in Table 3. The following differences exist in the types of terms, although it should be noted that the differences relate to TOTAL numbers of terms of any one type – many other terms involve different connections, and cannot be regarded as equivalent.

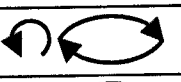
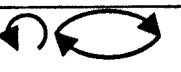
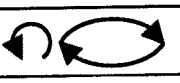
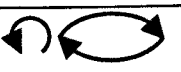
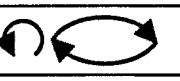
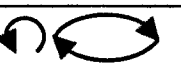
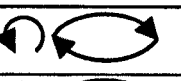
- 2 more paired 2 loops in System (a) than in System (b)
- 2 off 4- loops in System (a); none in System (b)
- 4 more (1-loop + 2 x 2-loop) terms in System (a) than in System (b)
- 4 off (1-loop + 4-loop) terms in System (a); none in System (b)
- 2 more (2 x 1-loop + 2 x 2-loop) terms in System (a) than in System (b)
- 2 off (2 x 1-loop + 4-loop) terms in System (a); none in System (b)

SYSTEM (a)		SYSTEM (b)	
Term	**Icon**	**Term**	**Icon**
λ^6		λ^6	
λ^5		λ^5	
(d_{11})		(d_{11})	
(d_{22})		(d_{22})	
(d_{33})		(d_{33})	
(d_{44})		(d_{44})	
(d_{55})		(d_{55})	
(d_{66})		(d_{66})	
λ^4		λ^4	
$(d_{55})(d_{66})$		$(d_{55})(d_{66})$	
$(d_{44})(d_{66})$		$(d_{44})(d_{66})$	
$(d_{44})(d_{55})$		$(d_{44})(d_{55})$	
$(d_{33})(d_{66})$		$(d_{33})(d_{66})$	
$(d_{33})(d_{55})$		$(d_{33})(d_{55})$	
$(d_{33})(d_{44})$		$(d_{33})(d_{44})$	
$(d_{11})(d_{66})$		$(d_{11})(d_{66})$	
$(d_{11})(d_{55})$		$(d_{11})(d_{55})$	

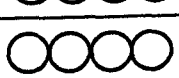
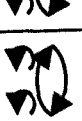
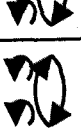
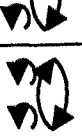
$(d_{11})(d_{44})$		$(d_{11})(d_{44})$	
$(d_{11})(d_{33})$		$(d_{11})(d_{33})$	
$(d_{22})(d_{66})$		$(d_{22})(d_{66})$	
$(d_{22})(d_{55})$		$(d_{22})(d_{55})$	
$(d_{22})(d_{44})$		$(d_{22})(d_{44})$	
$(d_{22})(d_{33})$		$(d_{22})(d_{33})$	
$(d_{11})(d_{22})$		$(d_{11})(d_{22})$	
$-(d_{23}\ d_{32})$			
$-(d_{25}\ d_{52})$			
$-(d_{12}\ d_{21})$		$-(d_{12}\ d_{21})$	
$-(d_{13}\ d_{31})$		$-(d_{13}\ d_{31})$	
$-(d_{56}\ d_{65})$		$-(d_{56}\ d_{65})$	
$-(d_{34}\ d_{43})$		$-(d_{34}\ d_{43})$	
$-(d_{15}\ d_{51})$		$-(d_{15}\ d_{51})$	
		$-(d_{14}\ d_{41})$	
		$-(d_{16}\ d_{61})$	

λ^3		λ^3	
$-(d_{11})(d_{33})(d_{66})$		$-(d_{11})(d_{33})(d_{66})$	
$-(d_{11})(d_{33})(d_{55})$		$-(d_{11})(d_{33})(d_{55})$	
$-(d_{11})(d_{33})(d_{44})$		$-(d_{11})(d_{33})(d_{44})$	
$-(d_{22})(d_{55})(d_{66})$		$-(d_{22})(d_{55})(d_{66})$	
$-(d_{22})(d_{44})(d_{66})$		$-(d_{22})(d_{44})(d_{66})$	
$-(d_{22})(d_{44})(d_{55})$		$-(d_{22})(d_{44})(d_{55})$	
$-(d_{22})(d_{33})(d_{66})$		$-(d_{22})(d_{33})(d_{66})$	
$-(d_{22})(d_{33})(d_{55})$		$-(d_{22})(d_{33})(d_{55})$	
$-(d_{22})(d_{33})(d_{44})$		$-(d_{22})(d_{33})(d_{44})$	
$-(d_{11})(d_{22})(d_{66})$		$-(d_{11})(d_{22})(d_{66})$	
$-(d_{11})(d_{22})(d_{55})$		$-(d_{11})(d_{22})(d_{55})$	

- $(d_{11})(d_{22})(d_{44})$		- $(d_{11})(d_{22})(d_{44})$	
- $(d_{11})(d_{22})(d_{33})$		- $(d_{11})(d_{22})(d_{33})$	
- $(d_{44})(d_{55})(d_{66})$		- $(d_{44})(d_{55})(d_{66})$	
- $(d_{33})(d_{55})(d_{66})$		- $(d_{33})(d_{55})(d_{66})$	
- $(d_{33})(d_{44})(d_{66})$		- $(d_{33})(d_{44})(d_{66})$	
- $(d_{33})(d_{44})(d_{55})$		- $(d_{33})(d_{44})(d_{55})$	
- $(d_{11})(d_{55})(d_{66})$		- $(d_{11})(d_{55})(d_{66})$	
- $(d_{11})(d_{44})(d_{66})$		- $(d_{11})(d_{44})(d_{66})$	
- $(d_{11})(d_{44})(d_{55})$		- $(d_{11})(d_{44})(d_{55})$	
$(d_{66})(d_{23}\ d_{32})$			
$(d_{55})(d_{23}\ d_{32})$			
$(d_{44})(d_{23}\ d_{32})$			
$(d_{11})(d_{23}\ d_{32})$			
$(d_{66})(d_{25}\ d_{52})$			
$(d_{11})(d_{25}\ d_{52})$			
$(d_{44})(d_{25}\ d_{52})$			
$(d_{33})(d_{25}\ d_{52})$			
$(d_{33})(d_{15}\ d_{51})$		$(d_{33})(d_{15}\ d_{51})$	
$(d_{22})(d_{15}\ d_{51})$		$(d_{22})(d_{15}\ d_{51})$	
$(d_{55})(d_{34}\ d_{43})$		$(d_{55})(d_{34}\ d_{43})$	
$(d_{44})(d_{56}\ d_{65})$		$(d_{44})(d_{56}\ d_{65})$	
$(d_{33})(d_{56}\ d_{65})$		$(d_{33})(d_{56}\ d_{65})$	
$(d_{66})(d_{34}\ d_{43})$		$(d_{66})(d_{34}\ d_{43})$	
$(d_{11})(d_{56}\ d_{65})$		$(d_{11})(d_{56}\ d_{65})$	
$(d_{11})(d_{34}\ d_{43})$		$(d_{11})(d_{34}\ d_{43})$	
$(d_{22})(d_{56}\ d_{65})$		$(d_{22})(d_{56}\ d_{65})$	
$(d_{22})(d_{34}\ d_{43})$		$(d_{22})(d_{34}\ d_{43})$	
$(d_{33})(d_{12}\ d_{21})$		$(d_{33})(d_{12}\ d_{21})$	

$(d_{44})(d_{12}\ d_{21})$		$(d_{44})(d_{12}\ d_{21})$	
$(d_{66})(d_{12}\ d_{21})$		$(d_{66})(d_{12}\ d_{21})$	
$(d_{55})(d_{12}\ d_{21})$		$(d_{55})(d_{12}\ d_{21})$	
$(d_{66})(d_{13}\ d_{31})$		$(d_{66})(d_{13}\ d_{31})$	
$(d_{55})(d_{13}\ d_{31})$		$(d_{55})(d_{13}\ d_{31})$	
$(d_{44})(d_{13}\ d_{31})$		$(d_{44})(d_{13}\ d_{31})$	
$(d_{22})(d_{13}\ d_{31})$		$(d_{22})(d_{13}\ d_{31})$	
$(d_{66})(d_{15}\ d_{51})$		$(d_{66})(d_{15}\ d_{51})$	
$(d_{44})(d_{15}\ d_{51})$		$(d_{44})(d_{15}\ d_{51})$	
		$(d_{22})(d_{14}\ d_{41})$	
		$(d_{33})(d_{14}\ d_{41})$	
		$(d_{33})(d_{16}\ d_{61})$	
		$(d_{44})(d_{16}\ d_{61})$	
		$(d_{55})(d_{14}\ d_{41})$	
		$(d_{55})(d_{16}\ d_{61})$	
		$(d_{66})(d_{14}\ d_{41})$	
		$(d_{22})(d_{16}\ d_{61})$	
$-(d_{12}\ d_{25}\ d_{51})$			
$-(d_{21}\ d_{52}\ d_{15})$			
$-(d_{12}\ d_{23}\ d_{31})$			
$-(d_{21}\ d_{32}\ d_{13})$			
		$-(d_{13}\ d_{34}\ d_{41})$	
		$-(d_{31}\ d_{43}\ d_{14})$	
		$-(d_{15}\ d_{56}\ d_{61})$	
		$-(d_{51}\ d_{65}\ d_{16})$	

λ^2		λ^2	
$(d_{11})(d_{44})(d_{55})(d_{66})$		$(d_{11})(d_{44})(d_{55})(d_{66})$	
$(d_{11})(d_{33})(d_{55})(d_{66})$		$(d_{11})(d_{33})(d_{55})(d_{66})$	

$(d_{11})(d_{33})(d_{44})(d_{66})$		$(d_{11})(d_{33})(d_{44})(d_{66})$	
$(d_{11})(d_{33})(d_{44})(d_{55})$		$(d_{11})(d_{33})(d_{44})(d_{55})$	
$(d_{22})(d_{44})(d_{55})(d_{66})$		$(d_{22})(d_{44})(d_{55})(d_{66})$	
$(d_{22})(d_{33})(d_{55})(d_{66})$		$(d_{22})(d_{33})(d_{55})(d_{66})$	
$(d_{22})(d_{33})(d_{44})(d_{66})$		$(d_{22})(d_{33})(d_{44})(d_{66})$	
$(d_{22})(d_{33})(d_{44})(d_{55})$		$(d_{22})(d_{33})(d_{44})(d_{55})$	
$(d_{11})(d_{22})(d_{55})(d_{66})$		$(d_{11})(d_{22})(d_{55})(d_{66})$	
$(d_{11})(d_{22})(d_{44})(d_{66})$		$(d_{11})(d_{22})(d_{44})(d_{66})$	
$(d_{11})(d_{22})(d_{44})(d_{55})$		$(d_{11})(d_{22})(d_{44})(d_{55})$	
$(d_{11})(d_{22})(d_{33})(d_{66})$		$(d_{11})(d_{22})(d_{33})(d_{66})$	
$(d_{11})(d_{22})(d_{33})(d_{55})$		$(d_{11})(d_{22})(d_{33})(d_{55})$	
$(d_{11})(d_{22})(d_{33})(d_{44})$		$(d_{11})(d_{22})(d_{33})(d_{44})$	
$(d_{33})(d_{44})(d_{55})(d_{66})$		$(d_{33})(d_{44})(d_{55})(d_{66})$	
$-(d_{55})(d_{66})(d_{13}\ d_{31})$			
$-(d_{33})(d_{44})(d_{25}\ d_{52})$			
$-(d_{11})(d_{66})(d_{25}\ d_{52})$			
$-(d_{11})(d_{66})(d_{23}\ d_{32})$			
$-(d_{44})(d_{66})(d_{25}\ d_{52})$			
$-(d_{11})(d_{55})(d_{23}\ d_{32})$			
$-(d_{44})(d_{55})(d_{23}\ d_{32})$			
$-(d_{33})(d_{66})(d_{25}\ d_{52})$			
$-(d_{44})(d_{66})(d_{23}\ d_{32})$			
$-(d_{11})(d_{44})(d_{25}\ d_{52})$			

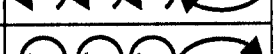
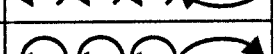
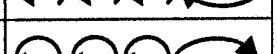
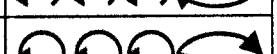
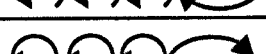
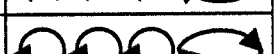
$-(d_{11})(d_{33})(d_{25}\ d_{52})$			
$-(d_{55})(d_{66})(d_{23}\ d_{32})$			
$-(d_{11})(d_{44})(d_{23}\ d_{32})$			
$-(d_{11})(d_{33})(d_{56}\ d_{65})$		$-(d_{11})(d_{33})(d_{56}\ d_{65})$	
$-(d_{11})(d_{66})(d_{34}\ d_{43})$		$-(d_{11})(d_{66})(d_{34}\ d_{43})$	
$-(d_{22})(d_{55})(d_{34}\ d_{43})$		$-(d_{22})(d_{55})(d_{34}\ d_{43})$	
$-(d_{22})(d_{44})(d_{56}\ d_{65})$		$-(d_{22})(d_{44})(d_{56}\ d_{65})$	
$-(d_{22})(d_{33})(d_{56}\ d_{65})$		$-(d_{22})(d_{33})(d_{56}\ d_{65})$	
$-(d_{22})(d_{66})(d_{34}\ d_{43})$		$-(d_{22})(d_{66})(d_{34}\ d_{43})$	
$-(d_{11})(d_{22})(d_{56}\ d_{65})$		$-(d_{11})(d_{22})(d_{56}\ d_{65})$	
$-(d_{11})(d_{22})(d_{34}\ d_{43})$		$-(d_{11})(d_{22})(d_{34}\ d_{43})$	
$-(d_{44})(d_{66})(d_{13}\ d_{31})$		$-(d_{44})(d_{66})(d_{13}\ d_{31})$	
$-(d_{44})(d_{55})(d_{13}\ d_{31})$		$-(d_{44})(d_{55})(d_{13}\ d_{31})$	
$-(d_{22})(d_{66})(d_{13}\ d_{31})$		$-(d_{22})(d_{66})(d_{13}\ d_{31})$	
$-(d_{22})(d_{55})(d_{13}\ d_{31})$		$-(d_{22})(d_{55})(d_{13}\ d_{31})$	
$-(d_{22})(d_{44})(d_{13}\ d_{31})$		$-(d_{22})(d_{44})(d_{13}\ d_{31})$	
$-(d_{33})(d_{66})(d_{15}\ d_{51})$		$-(d_{33})(d_{66})(d_{15}\ d_{51})$	
$-(d_{33})(d_{44})(d_{15}\ d_{51})$		$-(d_{33})(d_{44})(d_{15}\ d_{51})$	
$-(d_{22})(d_{66})(d_{15}\ d_{51})$		$-(d_{22})(d_{66})(d_{15}\ d_{51})$	

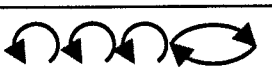
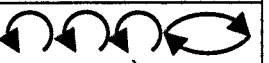
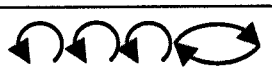
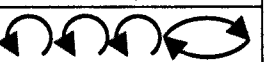
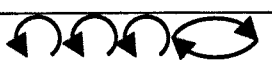
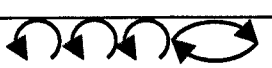
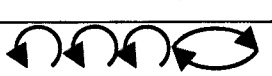
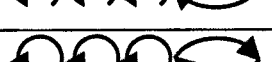
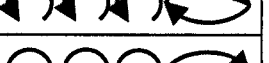
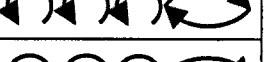
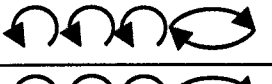
$-(d_{22})(d_{44})(d_{15}\ d_{51})$		$-(d_{22})(d_{44})(d_{15}\ d_{51})$	
$-(d_{22})(d_{33})(d_{15}\ d_{51})$		$-(d_{22})(d_{33})(d_{15}\ d_{51})$	
$-(d_{55})(d_{66})(d_{34}\ d_{43})$		$-(d_{55})(d_{66})(d_{34}\ d_{43})$	
$-(d_{55})(d_{66})(d_{12}\ d_{21})$		$-(d_{55})(d_{66})(d_{12}\ d_{21})$	
$-(d_{44})(d_{66})(d_{12}\ d_{21})$		$-(d_{44})(d_{66})(d_{12}\ d_{21})$	
$-(d_{44})(d_{55})(d_{12}\ d_{21})$		$-(d_{44})(d_{55})(d_{12}\ d_{21})$	
$-(d_{33})(d_{66})(d_{12}\ d_{21})$		$-(d_{33})(d_{66})(d_{12}\ d_{21})$	
$-(d_{33})(d_{55})(d_{12}\ d_{21})$		$-(d_{33})(d_{55})(d_{12}\ d_{21})$	
$-(d_{33})(d_{44})(d_{12}\ d_{21})$		$-(d_{33})(d_{44})(d_{12}\ d_{21})$	
$-(d_{33})(d_{44})(d_{56}\ d_{65})$		$-(d_{33})(d_{44})(d_{56}\ d_{65})$	
$-(d_{11})(d_{55})(d_{34}\ d_{43})$		$-(d_{11})(d_{55})(d_{34}\ d_{43})$	
$-(d_{11})(d_{44})(d_{56}\ d_{65})$		$-(d_{11})(d_{44})(d_{56}\ d_{65})$	
$-(d_{44})(d_{66})(d_{15}\ d_{51})$		$-(d_{44})(d_{66})(d_{15}\ d_{51})$	
		$-(d_{33})(d_{66})(d_{14}\ d_{41})$	
		$-(d_{33})(d_{55})(d_{14}\ d_{41})$	
		$-(d_{22})(d_{66})(d_{14}\ d_{41})$	
		$-(d_{22})(d_{55})(d_{14}\ d_{41})$	
		$-(d_{22})(d_{33})(d_{14}\ d_{41})$	
		$-(d_{55})(d_{66})(d_{14}\ d_{41})$	

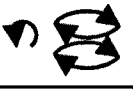
		$-(d_{22})(d_{33})(d_{16}\ d_{61})$	
		$-(d_{22})(d_{44})(d_{16}\ d_{61})$	
		$-(d_{22})(d_{55})(d_{16}\ d_{61})$	
		$-(d_{33})(d_{44})(d_{16}\ d_{61})$	
		$-(d_{33})(d_{55})(d_{16}\ d_{61})$	
		$-(d_{44})(d_{55})(d_{16}\ d_{61})$	
		$-(d_{11})(d_{33})(d_{56}\ d_{65})$	
$(d_{44})(d_{12}\ d_{25}\ d_{51})$			
$(d_{44})(d_{21}\ d_{52}\ d_{15})$			
$(d_{33})(d_{12}\ d_{25}\ d_{51})$			
$(d_{33})(d_{21}\ d_{52}\ d_{15})$			
$(d_{66})(d_{12}\ d_{23}\ d_{31})$			
$(d_{66})(d_{21}\ d_{32}\ d_{13})$			
$(d_{55})(d_{12}\ d_{23}\ d_{31})$			
$(d_{55})(d_{21}\ d_{32}\ d_{13})$			
$(d_{44})(d_{12}\ d_{23}\ d_{31})$			
$(d_{44})(d_{21}\ d_{32}\ d_{13})$			
$(d_{66})(d_{12}\ d_{25}\ d_{51})$			
$(d_{66})(d_{21}\ d_{52}\ d_{15})$			

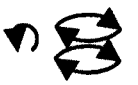
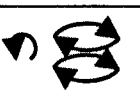
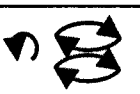
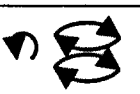
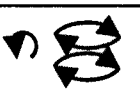
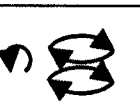
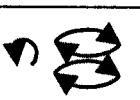
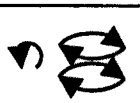
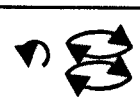
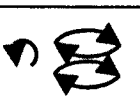
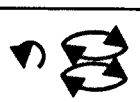
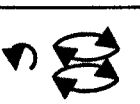
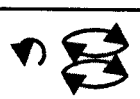
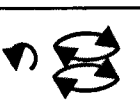
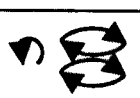
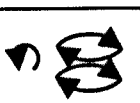
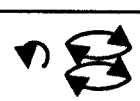
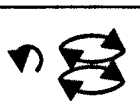
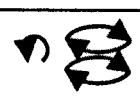
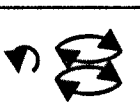
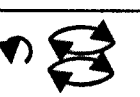
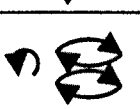
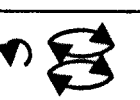
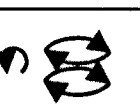
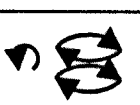
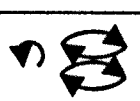
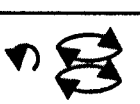
		$(d_{22})(d_{13}\ d_{34}\ d_{41})$	
		$(d_{22})(d_{31}\ d_{43}\ d_{14})$	
		$(d_{55})(d_{13}\ d_{34}\ d_{41})$	
		$(d_{55})(d_{31}\ d_{43}\ d_{14})$	
		$(d_{22})(d_{15}\ d_{56}\ d_{61})$	
		$(d_{22})(d_{51}\ d_{65}\ d_{16})$	
		$(d_{33})(d_{15}\ d_{56}\ d_{61})$	
		$(d_{33})(d_{51}\ d_{65}\ d_{16})$	
		$(d_{44})(d_{15}\ d_{56}\ d_{61})$	
		$(d_{44})(d_{51}\ d_{65}\ d_{16})$	
		$(d_{66})(d_{13}\ d_{34}\ d_{41})$	
		$(d_{66})(d_{31}\ d_{43}\ d_{14})$	
$(d_{23}\ d_{32})(d_{56}\ d_{65})$			
$(d_{15}\ d_{51})(d_{23}\ d_{32})$			
$(d_{13}\ d_{31})(d_{25}\ d_{52})$			
$(d_{34}\ d_{43})(d_{25}\ d_{52})$			
$(d_{34}\ d_{43})(d_{15}\ d_{51})$			
$(d_{34}\ d_{43})(d_{56}\ d_{65})$		$(d_{34}\ d_{43})(d_{56}\ d_{65})$	
$(d_{12}\ d_{21})(d_{56}\ d_{65})$		$(d_{12}\ d_{21})(d_{56}\ d_{65})$	
$(d_{12}\ d_{21})(d_{34}\ d_{43})$		$(d_{12}\ d_{21})(d_{34}\ d_{43})$	
$(d_{13}\ d_{31})(d_{56}\ d_{65})$		$(d_{13}\ d_{31})(d_{56}\ d_{65})$	
		$(d_{14}\ d_{41})(d_{56}\ d_{65})$	

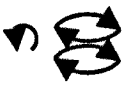
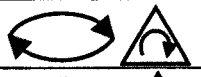
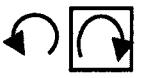
		$(d_{15}\ d_{51})(d_{34}\ d_{43})$	
		$(d_{16}\ d_{61})(d_{34}\ d_{43})$	
$-(d_{23}\ d_{31}\ d_{15}\ d_{52})$			
$-(d_{32}\ d_{13}\ d_{51}\ d_{25})$			

λ		λ	
$-(d_{11})(d_{33})(d_{44})(d_{55})(d_{66})$		$-(d_{11})(d_{33})(d_{44})(d_{55})(d_{66})$	
$-(d_{22})(d_{33})(d_{44})(d_{55})(d_{66})$		$-(d_{22})(d_{33})(d_{44})(d_{55})(d_{66})$	
$-(d_{11})(d_{22})(d_{44})(d_{55})(d_{66})$		$-(d_{11})(d_{22})(d_{44})(d_{55})(d_{66})$	
$-(d_{11})(d_{22})(d_{33})(d_{55})(d_{66})$		$-(d_{11})(d_{22})(d_{33})(d_{55})(d_{66})$	
$-(d_{11})(d_{22})(d_{33})(d_{44})(d_{66})$		$-(d_{11})(d_{22})(d_{33})(d_{44})(d_{66})$	
$-(d_{11})(d_{22})(d_{33})(d_{44})(d_{55})$		$-(d_{11})(d_{22})(d_{33})(d_{44})(d_{55})$	
$(d_{33})(d_{44})(d_{66})(d_{25}\ d_{52})$			
$(d_{11})(d_{33})(d_{44})(d_{25}\ d_{52})$			
$(d_{11})(d_{55})(d_{66})(d_{23}\ d_{32})$			
$(d_{11})(d_{33})(d_{66})(d_{25}\ d_{52})$			
$(d_{11})(d_{44})(d_{55})(d_{23}\ d_{32})$			
$(d_{11})(d_{44})(d_{66})(d_{23}\ d_{32})$			
$(d_{44})(d_{55})(d_{66})(d_{23}\ d_{32})$			
$(d_{11})(d_{44})(d_{66})(d_{25}\ d_{52})$			
$(d_{11})(d_{33})(d_{44})(d_{56}\ d_{65})$		$(d_{11})(d_{33})(d_{44})(d_{56}\ d_{65})$	
$(d_{22})(d_{55})(d_{66})(d_{34}\ d_{43})$		$(d_{22})(d_{55})(d_{66})(d_{34}\ d_{43})$	
$(d_{11})(d_{22})(d_{66})(d_{34}\ d_{43})$		$(d_{11})(d_{22})(d_{66})(d_{34}\ d_{43})$	
$(d_{22})(d_{33})(d_{44})(d_{56}\ d_{65})$		$(d_{22})(d_{33})(d_{44})(d_{56}\ d_{65})$	
$(d_{11})(d_{22})(d_{55})(d_{34}\ d_{43})$		$(d_{11})(d_{22})(d_{55})(d_{34}\ d_{43})$	
$(d_{11})(d_{22})(d_{44})(d_{56}\ d_{65})$		$(d_{11})(d_{22})(d_{44})(d_{56}\ d_{65})$	
$(d_{11})(d_{22})(d_{33})(d_{56}\ d_{65})$		$(d_{11})(d_{22})(d_{33})(d_{56}\ d_{65})$	
$(d_{44})(d_{55})(d_{66})(d_{13}\ d_{31})$		$(d_{44})(d_{55})(d_{66})(d_{13}\ d_{31})$	
$(d_{22})(d_{55})(d_{66})(d_{13}\ d_{31})$		$(d_{22})(d_{55})(d_{66})(d_{13}\ d_{31})$	

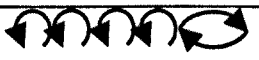
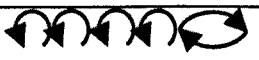
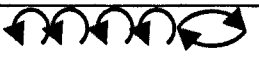
$(d_{22})(d_{44})(d_{66})(d_{13}\ d_{31})$		$(d_{22})(d_{44})(d_{66})(d_{13}\ d_{31})$	
$(d_{22})(d_{44})(d_{55})(d_{13}\ d_{31})$		$(d_{22})(d_{44})(d_{55})(d_{13}\ d_{31})$	
$(d_{33})(d_{44})(d_{66})(d_{15}\ d_{51})$		$(d_{33})(d_{44})(d_{66})(d_{15}\ d_{51})$	
$(d_{22})(d_{44})(d_{66})(d_{15}\ d_{51})$		$(d_{22})(d_{44})(d_{66})(d_{15}\ d_{51})$	
$(d_{22})(d_{33})(d_{66})(d_{15}\ d_{51})$		$(d_{22})(d_{33})(d_{66})(d_{15}\ d_{51})$	
$(d_{22})(d_{33})(d_{44})(d_{15}\ d_{51})$		$(d_{22})(d_{33})(d_{44})(d_{15}\ d_{51})$	
$(d_{44})(d_{55})(d_{66})(d_{12}\ d_{21})$		$(d_{44})(d_{55})(d_{66})(d_{12}\ d_{21})$	
$(d_{33})(d_{55})(d_{66})(d_{12}\ d_{21})$		$(d_{33})(d_{55})(d_{66})(d_{12}\ d_{21})$	
$(d_{33})(d_{44})(d_{66})(d_{12}\ d_{21})$		$(d_{33})(d_{44})(d_{66})(d_{12}\ d_{21})$	
$(d_{33})(d_{44})(d_{55})(d_{12}\ d_{21})$		$(d_{33})(d_{44})(d_{55})(d_{12}\ d_{21})$	
$(d_{11})(d_{55})(d_{66})(d_{34}\ d_{43})$		$(d_{11})(d_{55})(d_{66})(d_{34}\ d_{43})$	
		$(d_{33})(d_{55})(d_{66})(d_{14}\ d_{41})$	
		$(d_{22})(d_{55})(d_{66})(d_{14}\ d_{41})$	
		$(d_{22})(d_{33})(d_{66})(d_{14}\ d_{41})$	
		$(d_{22})(d_{33})(d_{55})(d_{14}\ d_{41})$	
		$(d_{22})(d_{33})(d_{44})(d_{16}\ d_{61})$	
		$(d_{22})(d_{33})(d_{55})(d_{16}\ d_{61})$	
		$(d_{22})(d_{44})(d_{55})(d_{16}\ d_{61})$	
		$(d_{33})(d_{44})(d_{55})(d_{16}\ d_{61})$	
$-(d_{55})(d_{66})(d_{12}\ d_{23}\ d_{31})$			
$-(d_{55})(d_{66})(d_{21}\ d_{32}\ d_{13})$			
$-(d_{44})(d_{66})(d_{12}\ d_{23}\ d_{31})$			
$-(d_{44})(d_{66})(d_{21}\ d_{32}\ d_{13})$			
$-(d_{44})(d_{55})(d_{12}\ d_{23}\ d_{31})$			
$-(d_{44})(d_{55})(d_{21}\ d_{32}\ d_{13})$			

- $(d_{44})(d_{66})(d_{12}\ d_{25}\ d_{51})$			
- $(d_{44})(d_{66})(d_{21}\ d_{52}\ d_{15})$			
- $(d_{33})(d_{66})(d_{12}\ d_{25}\ d_{51})$			
- $(d_{33})(d_{66})(d_{21}\ d_{52}\ d_{15})$			
- $(d_{33})(d_{44})(d_{12}\ d_{25}\ d_{51})$			
- $(d_{33})(d_{44})(d_{21}\ d_{52}\ d_{15})$			
		- $(d_{55})(d_{66})(d_{13}\ d_{34}\ d_{41})$	
		- $(d_{55})(d_{66})(d_{31}\ d_{43}\ d_{14})$	
		- $(d_{22})(d_{33})(d_{15}\ d_{56}\ d_{61})$	
		- $(d_{22})(d_{33})(d_{51}\ d_{65}\ d_{16})$	
		- $(d_{22})(d_{44})(d_{15}\ d_{56}\ d_{61})$	
		- $(d_{22})(d_{44})(d_{51}\ d_{65}\ d_{16})$	
		- $(d_{22})(d_{66})(d_{13}\ d_{34}\ d_{41})$	
		- $(d_{22})(d_{66})(d_{31}\ d_{43}\ d_{14})$	
		- $(d_{22})(d_{55})(d_{13}\ d_{34}\ d_{41})$	
		- $(d_{22})(d_{55})(d_{13}\ d_{34}\ d_{41})$	
		- $(d_{33})(d_{44})(d_{15}\ d_{56}\ d_{61})$	
		- $(d_{33})(d_{44})(d_{51}\ d_{65}\ d_{16})$	
- $(d_{44})(d_{15}\ d_{51})(d_{23}\ d_{32})$			

- $(d_{66})(d_{15}\ d_{51})(d_{23}\ d_{32})$			
- $(d_{66})(d_{25}\ d_{52})(d_{34}\ d_{43})$			
- $(d_{11})(d_{25}\ d_{52})(d_{34}\ d_{43})$			
- $(d_{66})(d_{13}\ d_{31})(d_{25}\ d_{52})$			
- $(d_{44})(d_{23}\ d_{32})(d_{56}\ d_{65})$			
- $(d_{11})(d_{23}\ d_{32})(d_{56}\ d_{65})$			
- $(d_{44})(d_{13}\ d_{31})(d_{25}\ d_{52})$			
- $(d_{22})(d_{34}\ d_{43})(d_{56}\ d_{65})$		- $(d_{22})(d_{34}\ d_{43})(d_{56}\ d_{65})$	
- $(d_{44})(d_{13}\ d_{31})(d_{56}\ d_{65})$		- $(d_{44})(d_{13}\ d_{31})(d_{56}\ d_{65})$	
- $(d_{22})(d_{13}\ d_{31})(d_{56}\ d_{65})$		- $(d_{22})(d_{13}\ d_{31})(d_{56}\ d_{65})$	
- $(d_{66})(d_{15}\ d_{51})(d_{34}\ d_{43})$		- $(d_{66})(d_{15}\ d_{51})(d_{34}\ d_{43})$	
- $(d_{22})(d_{15}\ d_{51})(d_{34}\ d_{43})$		- $(d_{22})(d_{15}\ d_{51})(d_{34}\ d_{43})$	
- $(d_{66})(d_{12}\ d_{21})(d_{34}\ d_{43})$		- $(d_{66})(d_{12}\ d_{21})(d_{34}\ d_{43})$	
- $(d_{55})(d_{12}\ d_{21})(d_{34}\ d_{43})$		- $(d_{55})(d_{12}\ d_{21})(d_{34}\ d_{43})$	
- $(d_{44})(d_{12}\ d_{21})(d_{56}\ d_{65})$		- $(d_{44})(d_{12}\ d_{21})(d_{56}\ d_{65})$	
- $(d_{33})(d_{12}\ d_{21})(d_{56}\ d_{65})$		- $(d_{33})(d_{12}\ d_{21})(d_{56}\ d_{65})$	
- $(d_{11})(d_{34}\ d_{43})(d_{56}\ d_{65})$		- $(d_{11})(d_{34}\ d_{43})(d_{56}\ d_{65})$	
		- $(d_{33})(d_{14}\ d_{41})(d_{56}\ d_{65})$	
		- $(d_{22})(d_{14}\ d_{41})(d_{56}\ d_{65})$	

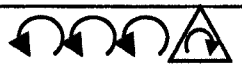
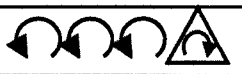
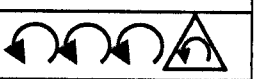
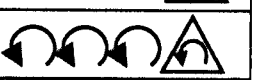
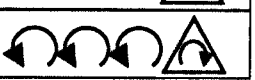
		$-(d_{22})(d_{34}\ d_{43})(d_{16}\ d_{61})$	
		$-(d_{55})(d_{34}\ d_{43})(d_{16}\ d_{61})$	
$(d_{34}\ d_{43})(d_{12}\ d_{25}\ d_{51})$			
$(d_{34}\ d_{43})(d_{21}\ d_{52}\ d_{15})$			
$(d_{56}\ d_{65})(d_{12}\ d_{23}\ d_{31})$			
$(d_{56}\ d_{65})(d_{21}\ d_{32}\ d_{13})$			
		$(d_{34}\ d_{43})(d_{15}\ d_{56}\ d_{61})$	
		$(d_{34}\ d_{43})(d_{51}\ d_{65}\ d_{16})$	
		$(d_{56}\ d_{65})(d_{13}\ d_{34}\ d_{41})$	
		$(d_{56}\ d_{65})(d_{31}\ d_{43}\ d_{14})$	
$(d_{66})(d_{13}\ d_{32}\ d_{25}\ d_{51})$			
$(d_{66})(d_{31}\ d_{23}\ d_{52}\ d_{15})$			
$(d_{44})(d_{13}\ d_{32}\ d_{25}\ d_{51})$			
$(d_{44})(d_{31}\ d_{23}\ d_{52}\ d_{15})$			

Constant terms		**Constant terms**	
$(d_{11})(d_{22})(d_{33})(d_{44})(d_{55})(d_{66})$		$(d_{11})(d_{22})(d_{33})(d_{44})(d_{55})(d_{66})$	

$-(d_{11})(d_{33})(d_{44})(d_{66})(d_{25}\ d_{52})$			
$-(d_{11})(d_{44})(d_{55})(d_{66})(d_{23}\ d_{32})$			
$-(d_{11})(d_{22})(d_{55})(d_{66})(d_{34}\ d_{43})$		$-(d_{11})(d_{22})(d_{55})(d_{66})(d_{34}\ d_{43})$	
$-(d_{11})(d_{22})(d_{33})(d_{44})(d_{56}\ d_{65})$		$-(d_{11})(d_{22})(d_{33})(d_{44})(d_{56}\ d_{65})$	
$-(d_{22})(d_{44})(d_{55})(d_{66})(d_{13}\ d_{31})$		$-(d_{22})(d_{44})(d_{55})(d_{66})(d_{13}\ d_{31})$	
$-(d_{22})(d_{33})(d_{44})(d_{66})(d_{15}\ d_{51})$		$-(d_{22})(d_{33})(d_{44})(d_{66})(d_{15}\ d_{51})$	
$-(d_{33})(d_{44})(d_{55})(d_{66})(d_{12}\ d_{21})$		$-(d_{33})(d_{44})(d_{55})(d_{66})(d_{12}\ d_{21})$	
		$-(d_{22})(d_{33})(d_{55})(d_{66})(d_{14}\ d_{41})$	
		$-(d_{22})(d_{33})(d_{44})(d_{55})(d_{16}\ d_{61})$	

$(d_{44})(d_{66})(d_{15}\ d_{51})(d_{23}\ d_{32})$			
$(d_{11})(d_{66})(d_{25}\ d_{52})(d_{34}\ d_{43})$			
$(d_{11})(d_{44})(d_{23}\ d_{32})(d_{56}\ d_{65})$			
$(d_{44})(d_{66})(d_{13}\ d_{31})(d_{25}\ d_{52})$			
$(d_{11})(d_{22})(d_{34}\ d_{43})(d_{56}\ d_{65})$		$(d_{11})(d_{22})(d_{34}\ d_{43})(d_{56}\ d_{65})$	
$(d_{22})(d_{44})(d_{13}\ d_{31})(d_{56}\ d_{65})$		$(d_{22})(d_{44})(d_{13}\ d_{31})(d_{56}\ d_{65})$	
$(d_{22})(d_{66})(d_{15}\ d_{51})(d_{34}\ d_{43})$		$(d_{22})(d_{66})(d_{15}\ d_{51})(d_{34}\ d_{43})$	
$(d_{55})(d_{66})(d_{12}\ d_{21})(d_{34}\ d_{43})$		$(d_{55})(d_{66})(d_{12}\ d_{21})(d_{34}\ d_{43})$	
$(d_{33})(d_{44})(d_{12}\ d_{21})(d_{56}\ d_{65})$		$(d_{33})(d_{44})(d_{12}\ d_{21})(d_{56}\ d_{65})$	
		$(d_{22})(d_{33})(d_{14}\ d_{41})(d_{56}\ d_{65})$	
		$(d_{22})(d_{55})(d_{34}\ d_{43})(d_{16}\ d_{61})$	

$-(d_{12}\ d_{21})(d_{34}\ d_{43})(d_{56}\ d_{65})$		$-(d_{12}\ d_{21})(d_{34}\ d_{43})(d_{56}\ d_{65})$	

$(d_{33})(d_{44})(d_{66})(d_{12}\ d_{25}\ d_{51})$			
$(d_{33})(d_{44})(d_{66})(d_{21}\ d_{52}\ d_{15})$			
$(d_{44})(d_{55})(d_{66})(d_{12}\ d_{23}\ d_{31})$			
$(d_{44})(d_{55})(d_{66})(d_{21}\ d_{32}\ d_{13})$			
		$(d_{22})(d_{33})(d_{44})(d_{15}\ d_{56}\ d_{61})$	
		$(d_{22})(d_{33})(d_{44})(d_{51}\ d_{65}\ d_{16})$	
		$(d_{22})(d_{55})(d_{66})(d_{13}\ d_{34}\ d_{41})$	
		$(d_{22})(d_{55})(d_{66})(d_{31}\ d_{43}\ d_{14})$	

$-(d_{44})(d_{56}\ d_{65})(d_{12}\ d_{23}\ d_{31})$			
$-(d_{44})(d_{56}\ d_{65})(d_{21}\ d_{32}\ d_{13})$			
$-(d_{66})(d_{34}\ d_{43})(d_{12}\ d_{25}\ d_{51})$			
$-(d_{66})(d_{34}\ d_{43})(d_{21}\ d_{52}\ d_{15})$			
		$-(d_{22})(d_{34}\ d_{43})(d_{15}\ d_{56}\ d_{61})$	
		$-(d_{22})(d_{34}\ d_{43})(d_{51}\ d_{65}\ d_{16})$	

		- $(d_{22})(d_{56}\ d_{65}\ )(d_{13}\ d_{34}\ d_{41})$	
		- $(d_{22})(d_{56}\ d_{65}\ )(d_{31}\ d_{43}\ d_{14})$	
- $(d_{44})(d_{66})(d_{13}\ d_{32}\ d_{25}\ d_{51})$			
- $(d_{44})(d_{66})(d_{31}\ d_{23}\ d_{52}\ d_{15})$			

Table 9.3:Comparison of CP Terms of the two Co-Spectral Graphs Considered, based on **C**

9.4 Commentary

It was shown earlier in this chapter that cospectral systems can have very different characteristics, as shown by the simple example of mobilities. The significance of Tables 9.2 and 9.3 is that the concepts proposed in Chapter 8 for the representation of system fault paths using fault classes and fault categories can be reliably applied, knowing that any occurrence of cospectral graphs will be catered for by the method.

However, it is important to understand that the foregoing illustrations of the differences that can be identified between cospectral systems apply **only in the case of non-isomorphic cospectral systems**. Consider the 2D cospectral graphs discussed in Section 6.5.2:

- Fault path 1: System 2.1.2 → System 2.1.3
- Fault Path 2: System 2.1.4 → System 2.1.5 → System 2.1.6
- Fault Path 3: System 2.1.7 → System 2.1.8 → System 2.1.9 → System 2.1.10

Systems with the same colour coding are cospectral, but the isomorphism situation needs to be examined also, as shown in Table 9.4, below. As stated in 6.5.2, there is the possibility for two alternative views of isomorphism – **adjacency isomorphism** and **constraints isomorphism**. This depends on whether an Adjacency Matrix, **A**, or Constraints Matrix, **C**, view of isomorphism is taken – that is to say whether degrees of freedom are taken into account in deciding what vertex connectivity exists.

Table 9.4 shows that all the 2D systems considered are adjacency isomorphic, only groups of them are constraints isomorphic, and these same constraints isomorphic groups are also cospectral. This latter relationship is to be expected since the characteristic polynomials and constraints

isomorphism are both derived from **C**. As has been pointed out before, it is necessary for all isomorphic systems to be cospectral, but not necessarily the case that cospectral systems are isomorphic. In this situation, the symbolic determinants approach does not operate, as can be seen from the table, which shows that all the symbolic determinants are the same.

System	Characteristic Polynomial	Symbolic Determinant	Cospectral?	Adjacency Isomorphism?	Constraints Isomorphism?
2.1.2	$\lambda^2 - 12\lambda + 11$	$(d_{11}.d_{22}) - (d_{12}.d_{21})$	2.1.2, 2.1.5 and 2.1.9 are cospectral	All systems in table are adjacency isomorphic	2.1.2, 2.1.5 and 2.1.9 are constraints isomorphic
2.1.5	$\lambda^2 - 12\lambda + 11$	$(d_{11}.d_{22}) - (d_{12}.d_{21})$			
2.1.9	$\lambda^2 - 12\lambda + 11$	$(d_{11}.d_{22}) - (d_{12}.d_{21})$			
2.1.3	$\lambda^2 - 12\lambda$	$(d_{11}.d_{22}) - (d_{12}.d_{21})$	2.1.3, 2.1.6 and 2.1.10 are cospectral	All systems in table are adjacency isomorphic	2.1.3, 2.1.6 and 2.1.10 are constraints isomorphic
2.1.6	$\lambda^2 - 12\lambda$	$(d_{11}.d_{22}) - (d_{12}.d_{21})$			
2.1.10	$\lambda^2 - 12\lambda$	$(d_{11}.d_{22}) - (d_{12}.d_{21})$			
2.1.4	$\lambda^2 - 12\lambda + 20$	$(d_{11}.d_{22}) - (d_{12}.d_{21})$	2.1.4, and 2.1.8 are cospectral	All systems in table are adjacency isomorphic	2.1.4, and 2.1.8 are constraints isomorphic
2.1.8	$\lambda^2 - 12\lambda + 20$	$(d_{11}.d_{22}) - (d_{12}.d_{21})$			

Table 9.4:Comparison of 2D Co-Spectral and Isomorphic Graphs

The following conclusions can therefore be drawn:

- the method of analysing Characteristic Polynomials by means of their constituent terms is a useful way of discriminating between co-spectral graphs, but is of more limited use where isomorphism is present as well.
- Provided that care is taken to avoid potential isomorphism problems, the symbolic determinant approach is valid for examining cospectral systems, and the validity of the fault class method discussed in Chapter 8 is reinforced by this work on cospectral systems.
- A final point to note is that the example of the 2D systems discussed above shows that it is necessary to apply a consistent adjacency- or constraints-based approach in dealing with system representations, if conflicting, or at least, confusing, results are to be avoided. If a decision is taken to mix the two, care needs to be taken to ensure that conflicts of definition do not arise.